UNIVERSIDAD AUTONÓMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de la Telecomunicación

TRABAJO FIN DE GRADO

Segmentación de tráfico en Internet mediante la clasificación de parámetros estadísticos

Alberto Ruiz Santos
Tutor: Luis de Pedro Sánchez
Ponente: Jorge Enrique López de Vergara

2019

Segmentación de tráfico en Internet mediante la clasificación de parámetros estadísticos

AUTOR: Alberto Ruiz Santos
TUTOR: Luis de Pedro Sánchez

Computación y redes de alta prestaciones
Dpto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2019

Resumen (castellano)

El tráfico en Internet ha ido aumentando exponencialmente en la última década, y con ello los ataques en la red o ciberataques. Además, debido al desarrollo de nuevas técnicas de hacking, cada vez estos ataques son más difíciles de detectar. La importancia de desarrollar métodos que nos protejan de estos ataques es crítica, pero para ello hay que entender la forma que tiene y cómo afectan a las características del tráfico.

De una forma paralela a este aumento, también se han desarrollado varios modelos estadísticos y sistemas que tienen como objetivo analizar el tráfico de la red bajo ataque y ver que características tiene para poder proporcionar información y elaborar protocolos de actuación cuando se producen dichos ataques.

El análisis de los modelos estadísticos de tráfico en una red permite caracterizarlo, hasta tal punto de poder identificar ataques informáticos utilizando ajustes estadísticos de series temporales mediante la distribución α-estable.

El objetivo de este trabajo consiste en utilizar el modelo α-estable, aplicarlo a un conjunto de datos proporcionados por el autor del TFG de partida, que se encargó de analizar esta distribución y ver si hay alguna diferencia entre el tráfico normal y el tráfico de ataque.

Se van a analizar dos series, una formada por el tráfico real de una red y otra formada por tráfico sintético de ataque añadido a la serie real. Con ambas, se intentará hacer una especie de clusterización donde se podrá determinar si existen zonas dentro de un espacio de fases formados por estos parámetros que sean exclusivas del tráfico normal o del de ataque.

Palabras clave (Castellano)

Matlab, α-estable, k-means, clasificación, aprendizaje supervisado, denegación de servicio, modelo estadístico, ciberataques.

Abstract (English)

Internet traffic has been increasing exponentially in the last decade, and with it attacks on the Internet or cyber-attacks. In addition, due to the development of new hacking techniques, these attacks are increasingly more difficult to detect. The importance of developing methods that protect us from these attacks is critical, but first we must understand the shape it has and how they affect traffic characteristics.

Parallel to this increase, several statistical models and systems have been developed that aim to analyze the traffic of the network under attack and see what characteristics it has in order to provide information and develop action protocols when such attacks occur.

The analysis of statistical traffic models in a network allows us to characterize it, to the extent that we can identify computer attacks by using temporal series statistical adjustments through the α-stable distribution.

The objective of this work is to use the α-stable model, to apply it to a set of data provided by the author of the starting Bachelor Thesis, who has analyzed this distribution and figure out if there was any difference between normal traffic and traffic of attack.

Two series will be analyzed, one formed by the real traffic of a network and another one formed by synthetic traffic of attack added to the real traffic series. With both, we will try to do a kind of clustering to determine if there are zones within the space of phases formed by these parameters that are exclusive of normal or attack traffic.

Keywords (English)

Matlab, α-stable, k-means, classification, supervised learning, denial of service, statistical model, cyber-attacks.

Agradecimientos

Antes de empezar con este trabajo me gustaría dar las gracias al autor del TFG del que se parte, Eric Crusi Mozota, que nos ha proporcionado todos los datos necesarios para realizar este trabajo.

Además, también quiero agradecer a tanto a mi tutor como a mi ponente toda la ayuda que me han proporcionado, incluyendo todas las conferencias de Skype. A mis compañeros de clase que han estado conmigo, ayudándonos mutuamente a lo largo de todos estos años, en especial a Luis Aguilera, Alejandro Camacho, a mis amigos del instituto que también son de la carrera, Mario González y Raúl Arcos, y por último a mi familia que siempre ha estado ahí en los peores momentos.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

1 Introducción

En esta sección se expondrá cual ha sido la motivación de hacer este TFG, los objetivos, las fases en las que ha sido realizado y la estructura del documento.

1.1 Motivación

Durante estos últimos años ha habido un incremento del número de ciberataques por parte de diferentes organizaciones y cada vez a mayor escala. Es de vital importancia que las empresas que manejan nuestros datos personales y los suyos propios mantengan y garanticen la seguridad de estos datos. Uno de los ataques más comunes a este tipo de entidades son los ataques de denegación de servicio (DoS). Son tan comunes porque son relativamente fáciles de llevar a cabo y pueden tener consecuencias devastadoras si se logran hacer bien. Además, cada día surgen nuevos tipos de ciberataques que son muy difíciles de detectar.

El estudio de los ciberataques es de vital importancia ya que ayuda no sólo a detectarlos sino a prevenirlos. Gracias a la distribución α-estable, hemos podido ver cuando se produce el ataque y ver si se puede extraer un patrón que pueda ser utilizado para la detección de posibles ataques DoS.

Este trabajo consiste en desarrollar un algoritmo de clasificación que ayude a separar en regiones el tráfico de ataque y el tráfico normal, pudiendo distinguirlos de una manera clara.

1.2 Objetivos

El principal objetivo de este TFG es el análisis y estudio de los parámetros obtenidos del TFG previo del que se parte, poder detectar algún tipo de patrón e incluso ver si se puede aplicar un algoritmo de clasificación efectivo para así poder detectar futuros ataques DoS. Este proceso se ha dividido en:

- Obtención de los parámetros α-estable
- Representar estos parámetros
- Decidir qué representación se podía clasificar mejor
- Aplicar los algoritmos de clasificación
- Analizar los resultados

El siguiente diagrama de bloques muestra un resumen de todo lo realizado en este trabajo. Los módulos destacados como tridimensionales son los componentes de este TFG. Es importante destacar que los α-estables también se calcularon en el trabajo de partida, pero con diferentes tamaños de ventana.

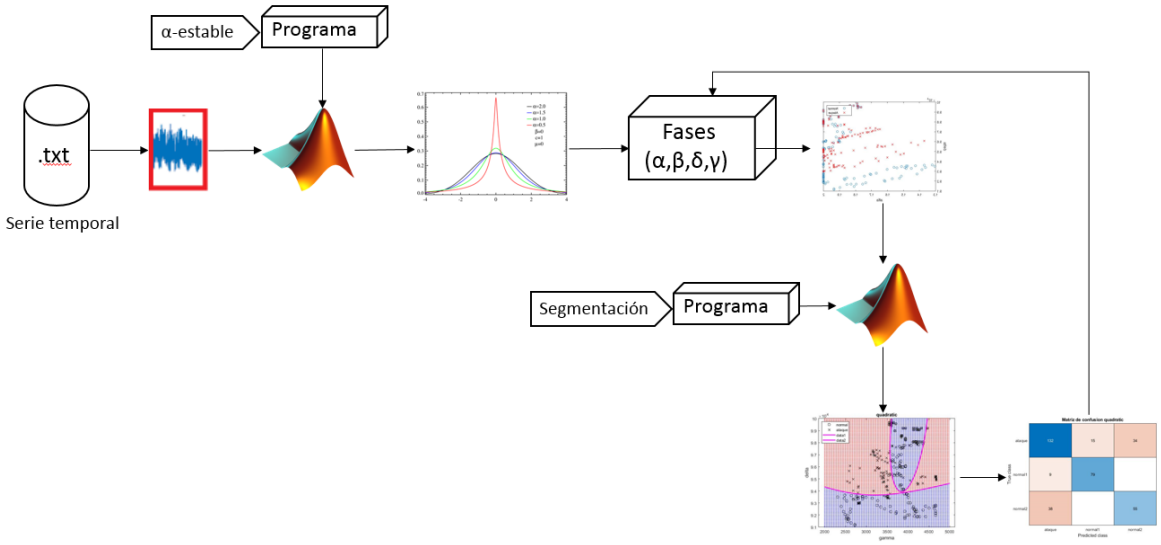


Figura 1-1: Diagrama de bloques

1.3 Fases de realización

En este punto se hará un resumen de todas las fases en las que se ha dividido este TFG, hasta llegar a todos los resultados extraídos. Estas fases son:

-Obtención de parámetros: a partir del código desarrollado en el TFG del que se parte se han calculado todos los parámetros α-estable de la serie de datos suministrada.

-Representación de parámetros: una vez que se han obtenido, se han representado todas las posibles combinaciones, incluso en 3D.

-Análisis de las representaciones: después de obtener todas las representaciones se ha analizado cual de todas las posibles son más adecuadas para la posterior aplicación de los algoritmos.

-Desarrollo de código: esta fase engloba todo el trabajo ya que para todas las etapas se ha ido escribiendo todo el código necesario para poder hacer todas las tareas. Además, se estudiaron todas las posibles herramientas y las propias funciones de Matlab relacionadas con el aprendizaje y la segmentación.

-Aplicación de algoritmos: cuando ya se sabía cuáles eran las funciones idóneas a aplicar se procedió al desarrollo del código que permitía la aplicación de los algoritmos que se querían probar y analizar.

-Parámetros estadísticos significativos: una vez que se han aplicado los algoritmos se ha procedido a analizar cuáles eran aquellos parámetros de la distribución α-estable que más influencia tenían a la hora de clasificar.

-Análisis: tras la aplicación de los algoritmos se han analizado todos los resultados de estos, incluyendo el error de estos algoritmos y la clasificación de los pares (α, δ) y (γ, δ) llegando a varias conclusiones que se expondrán en este trabajo.

A continuación de muestra el diagrama de Gantt en relación a este trabajo:

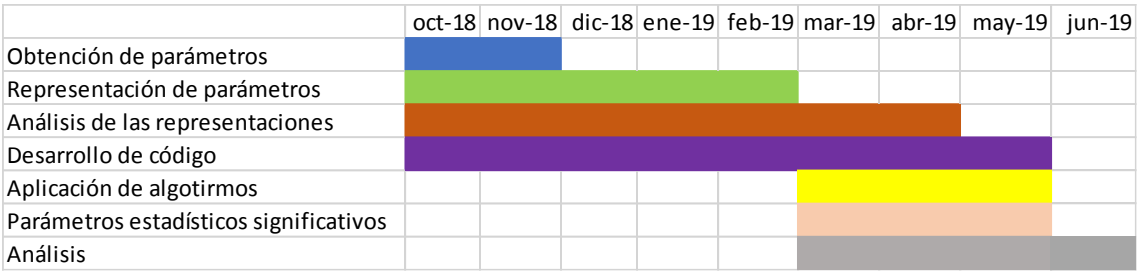


Figura 1-2: Diagrama de Gantt

1.4 Organización de la memoria

Esta memoria cuenta con los siguientes capítulos:

Capítulo 1: Introducción: como ya se ha hecho, en este capítulo se ha expuesto una introducción de todo el TFG.

Capítulo 2: Estado del arte: aquí se explicarán todas las ideas y conceptos clave para poder llegar a entender todo lo realizado en este trabajo.

Capítulo 3: Diseño: a lo largo de este capítulo se explicará toda la fase de diseño que se ha llevado a cabo para realizar el trabajo.

Capítulo 4: Desarrollo: a partir de lo expuesto en la etapa de diseño, en este apartado se explicará todas las etapas de desarrollo del trabajo.

Capítulo 5: Pruebas y resultados: en esta sección se hablará de todos los resultados obtenidos y de las diferentes variaciones que se han hecho.

Capítulo 6: Conclusiones y trabajo futuro: una vez obtenidos los resultados, en este capítulo se habla de las conclusiones a las que se ha llegado y los posibles trabajos que pueden derivar de este TFG.

2 Estado del arte

2.1 Introducción

En este capítulo se van a exponer todos los conocimientos teóricos básicos necesarios que están relacionados con este trabajo y que ayudan a la comprensión y al dominio de todos los conceptos expuestos en este trabajo.

2.2 Distribución alfa-estable

Una distribución es estable si es una combinación lineal de dos o mas copias independientes de una misma variable aleatoria con la misma función de probabilidad. Sean X_1 y X_2 dos copias independientes de una variable aleatoria X. X es estable si existen dos constantes a>0 y b>0 tales que la suma $aX_1 + bX_2$ tenga la misma distribución que cX + d:

$aX_1 + bX_2 = cX + d$, c>0

Debido a sus características y sus herramientas matemáticas, es el tipo de distribución que mejor se adapta al análisis de tráfico. La densidad de probabilidad de este tipo de distribuciones no puede expresarse mediante fórmulas elementales, pero su función característica sí, mediante la transformada de Fourier:

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \varphi(t)e^{-ixt}\,dt$$

En Matlab, la distribución que emplea es S (α, β, γ, δ_0; 0) cuya función característica es la siguiente:

$$E(e^{itX}) = \begin{cases} \exp\left(-\gamma^{\alpha}|t|^{\alpha}\left[1 + i\beta\,\mathrm{sign}(t)\tan\frac{\pi\alpha}{2}\left((\gamma|t|)^{1-\alpha} - 1\right)\right] + i\delta_0 t\right) & for\ \alpha \neq 1, \\ \exp\left(-\gamma|t|\left[1 + i\beta\,\mathrm{sign}(t)\frac{2}{\pi}\ln(\gamma|t|)\right] + i\delta_0 t\right) & for\ \alpha = 1 \end{cases}$$

Esta distribución recibe el nombre de α-estable de Lévy, en honor a Paul Lévy, que fue el primero en estudiar esta distribución.

Para el caso particular de la distribución α-estable los parámetros son los siguientes:

- Parámetro de estabilidad: α. Es el valor más importante ya que define como es de estable la función, es decir, la forma que tiene la curva. $0 < \alpha \leq 2$.
- Coeficiente de simetría: β. Determina cómo es de simétrica la función. $-1 \leq \beta \leq 1$.
- Parámetro de escala: γ. Cuanto mayor es este parámetro, más amplia es la distribución. En la figura viene dada como la letra 'c'. $\gamma > 0$.
- Parámetro de localización: δ. En la figura viene dada como 'μ'. $\delta \in \mathbb{R}$

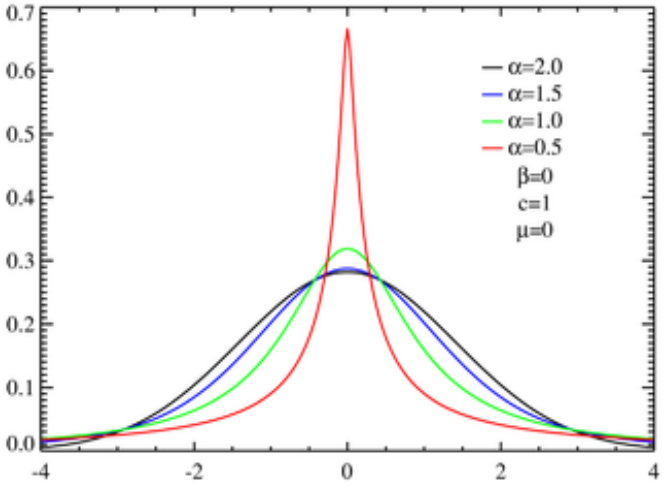


Figura 2-1: Distribución α-estable

2.3 Ataques informáticos

Un ataque informático o ciberataque es un intento de alterar, boicotear, eliminar para obtener acceso a un servicio sin autorización o de provocar daño a un sistema informático. Existen muchos tipos de ciberataques, pero se van a explicar los ataques de denegación de servicio (DoS) y los ataques de denegación de servicios distribuidos (DDoS):

- Ataque DoS: Es un tipo de ataque informático que se hace a una red con el objetivo de que un servicio o recurso sea inaccesible a los usuarios que lo solicitan. Provocan la pérdida de la conectividad a una red normalmente por la sobrecarga de los recursos informáticos del sistema atacado. Los ataques DoS se generan mediante la saturación de los puertos del servidor a través de múltiples flujos de datos, haciendo que este se sobrecargue y deje de funcionar. Esta técnica es utilizada por los hackers para dejar fuera de línea a servidores objetivo. Es un problema que ha ido creciendo estos últimos años por la facilidad de generar estos ataques y por la gran cantidad de equipos con fallos de seguridad que son aprovechados por los piratas informáticos.

- Ataque DDoS: es una variante mejorada del ataque DoS, que consiste en mandar una gran cantidad de flujos de información a un mismo destino desde varios puntos al mismo tiempo. Son de mayor escala que los ataques DoS ya que producen daños y colapsos mayores. Se suele llevar a cabo mediante una red de bots que mandan flujos de datos enormes haciendo que los servidores no puedan procesar esa cantidad de solicitudes y colapsen.

Existen varios métodos de ataque DoS:

- Inundación SYN: cuando un nodo se comunica con otro mediante TCP/IP, los paquetes que se envían están formados por una serie de datos junto con la solicitud real. Estos datos constituyen la cabecera de la petición. En la cabecera se encuentran los *Flags*. Estos *Flags* tienen diferentes usos, como iniciar una

conexión, cerrarla, reiniciar una conexión… Estas banderas se incluyen tanto en la petición del cliente como en la respuesta del servidor.

Este tipo de inundación manda un flujo de paquetes TCP con el Flag SYN activado, pero con direcciones IP de origen falsificadas que no existen. Por tanto, el servidor crea conexiones al responder a los paquetes TCP/SYN-ACK que nunca se van a cerrar, ya que se queda esperando a que el cliente le devuelva el paquete TCP/ACK. Esto consume recursos del servidor hasta que llegar un punto en el que se colapsa.

- Inundación ICMP: es una técnica que tiene como objetivo agotar el ancho de banda de la víctima, mediante el envío de un alto número de paquetes ICMP (ping) que son de tamaño considerable. El éxito de esta inundación depende del ancho de banda que tenga el usuario atacado, incluso puede llegar a fracasar si la máquina y la red de la víctima tienen la suficiente capacidad para procesar todos estos paquetes, la red se vería ralentizada, pero en ningún caso colapsaría la línea del usuario.

- Inundación SMURF: es una variante que amplifica el ataque ICMP. Está formado por tres partes: atacante, intermediario y la víctima (destacar que también el intermediario puede ser una segunda víctima). El atacante envía paquetes ICMP a una dirección IP de broadcast, usando como dirección IP origen la de la víctima. Esto provoca que la cantidad de respuestas varía en función de los equipos activos en la red broadcast que respondan a esa petición. Todas las respuestas van dirigidas a la víctima colapsando sus recursos de red.

Estos ataques pueden tener repercusiones a nivel global si son capaces de colapsar granjas de servidores y hacer que todo el tráfico global de internet se ralentice e incluso llegar a afectar a puntos clave de la red.

2.4 Algoritmos de clasificación

A lo largo de todo el desarrollo de este TFG se han utilizado numerosos algoritmos tanto de clasificación como de clusterización. En este punto se exponen brevemente cada uno de estos algoritmos.

- Aprendizaje supervisado: es una técnica que se usa para predecir el comportamiento de una serie de datos de entrada a partir de unos datos de entrenamiento. A este tipo de algoritmos se les da unos datos ya clasificados y lo que hacen es establecer una etiqueta de clase para cada dato. Una vez que se tienen esas etiquetas, es evaluar la precisión del modelo de predicción desarrollado con datos reales. En este TFG sólo se va a estudiar la clasificación y la posible segmentación que hacen estos algoritmos con los datos de entrenamiento suministrados, pero no se evaluará su funcionalidad en casos reales. En nuestro caso se va a utilizar el análisis discriminante lineal, cuadrático y por distancia Mahalanobis.

    - Fórmula de la distancia Mahalanobis desde un vector y a una distribución con media μ y covarianza Σ es:

$$d = \sqrt{(y-\mu)\sum^{-1}(y-\mu)'}.$$

- Aprendizaje no supervisado: es una técnica que se distingue del aprendizaje supervisado en que no parte de un conocimiento a priori de los datos de entrada, es decir, no hay un modelo de predicción ya hecho. Este aprendizaje trata al conjunto de datos de entrada como variables aleatorias, construyendo un modelo de densidad para el conjunto de objetos. El tipo de aprendizaje no supervisado que se va a aplicar a nuestro conjunto de datos es el conocido K-means. Es un método de agrupamiento que divide un conjunto de *m* observaciones en *c* grupos, en el que cada observación pertenece a un grupo o a otro en función de su distancia media a cada grupo. Se parte de *c* centroides que son generados aleatoriamente dentro del conjunto de datos. A cada punto del conjunto se le asigna el centroide más cercano. Por último, el centroide de cada grupo de recalcula en función de la distancia media a cada punto de ese grupo. Esto se hace un número de iteraciones que depende de las necesidades de cada caso.

2.5 Conclusiones

Gracias a la distribución α-estable es posible saber, tras el correspondiente análisis de datos, si, a posteriori, una compañía o un usuario corriente ha sufrido un ataque de denegación de servicio. Con los algoritmos de aprendizaje y clasificación se podría hacer una segmentación de los parámetros α-estable que podrían servir para detectar, clasificar o estudiar el comportamiento de ciertos tipos de ciberataques.

3 Diseño

3.1 Introducción

En este capítulo se explica todo el proceso de diseño que se ha seguido en este trabajo. Además, en cada sección se explicará el motivo de las decisiones tomadas y las repercusiones que tienen. Los temas tratados son:

- Ventanas temporales
- Ajuste estadístico
- Espacios de fases
- Segmentación
- Metodología

3.2 Ventanas temporales

Antes de explicar la utilidad de las ventanas temporales deslizantes es necesario saber que series temporales se van a utilizar para el análisis y el cálculo de parámetros α-estables. Estas dos series fueron proporcionadas por el autor del TFG de partida. Había una serie original que contenía los datos de una semana de tráfico real que fue descargada de la Universidad de Granada. De ese mismo sitio también fue descargada una serie de tráfico sintético que duraba 2 minutos y representaba un ataque de denegación de servicio. Para generar la nueva serie temporal, se mezclaron el tráfico real con el tráfico sintético, dando lugar a una nueva serie temporal que contenía el ataque.

Para la estimación de los parámetros se ha utilizado una ventana deslizante sobre ambas series temporales tanto en bits como en paquetes por segundo. El ajuste de la distribución α-estable se realiza para los datos que están incluidos dentro de la propia ventana, que se va desplazando de segundo en segundo. Este ajuste se realiza en Matlab, cuyo procedimiento ser verá en el siguiente capítulo. En la siguiente figura se muestra un ejemplo de una ventana de 15 minutos sobre la representación de los bits por segundo de la serie correspondiente al tráfico real.

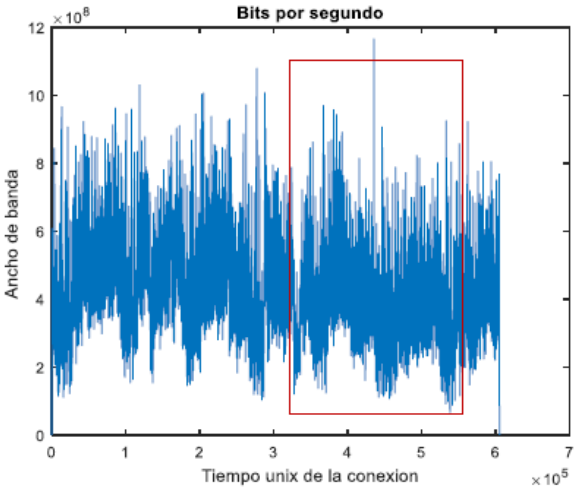


Figura 3-1: ejemplo de ventana deslizante

En el TFG de partida se eligieron ventanas de 2 y 15 minutos. Para nuestro trabajo ha sido necesario emplear ventanas temporales de otros tamaños. Estas ventanas están centradas en la zona de 2 minutos que dura el ataque. Sin embargo, un número elevado de puntos haría casi imposible una clasificación ya que cuanto mayor es el tamaño de la ventana más puntos que no son de la zona de ataque se tienen en cuenta para calcular los parámetros α-estables.

Las ventanas que se propusieron fueron las de 5, 10 y 15 minutos y se estudio cuál era la más idónea para hacer la clasificación. Estas ventanas solo se aplican en la zona del ataque, ya que no interesa el resto de la serie porque el tráfico es igual El número de puntos incrementa según aumentamos el tamaño de la ventana por tanto había que escoger una ventana que no tuviera una gran cantidad de puntos, pero los suficientes como para que la clasificación tuviera sentido y se pudieran sacar conclusiones. Para contabilizar el número de puntos por ventana el cálculo es muy sencillo, puesto que se tiene una muestra por segundo; basta con restar los segundos que dura la ventana y los que dura el ataque. Así para una ventana de 10 minutos habría 480 puntos.

En las ventanas de 10 y de 15 minutos observamos que la distribución de los puntos no seguía ningún tipo de patrón y no se diferenciaban zonas ni de tráfico de ataque ni de tráfico normal, por lo que intentar clasificar y segmentar los parámetros era inviable. Sin embargo, se observó que, con ventanas de 5 minutos, en algunos pares de parámetros sí que podíamos diferenciar claramente las zonas de ataque y las zonas de tráfico normal cosa que en ventanas de mayor tamaño no se podía. **Por tanto, todos los cálculos y clasificaciones se han hecho con ventanas de 5 minutos.**

Las figuras siguientes muestran la diferencia explicada anteriormente. En el caso de la ventana de 15 minutos se puede observar que hay muchos más puntos que se solapan tanto de tráfico de ataque como de tráfico normal lo que provocaría que la clasificación tuviera mucho error. Sin embargo, utilizando una ventana de 5 minutos podemos ver dos regiones separadas entre los de tipos de tráfico.

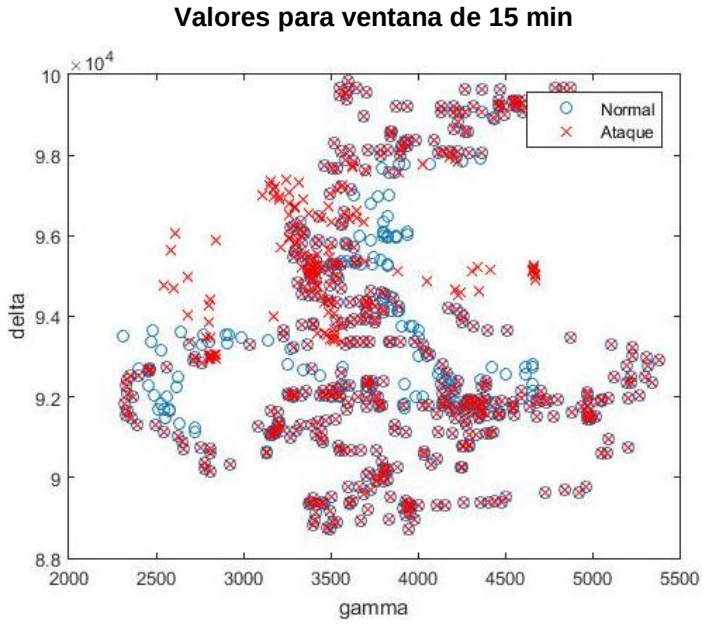


Figura 3-2: gamma-delta 15 min

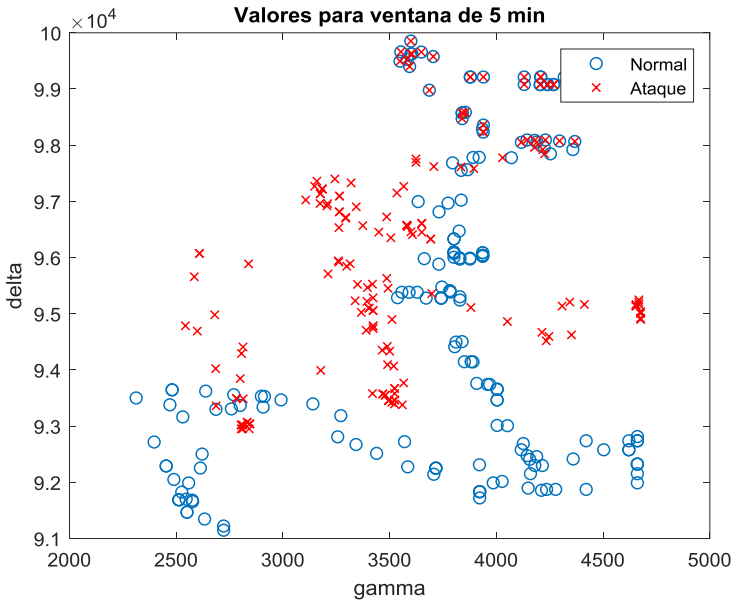


Figura 3-3: gamma-delta 5 min

3.3 Ajuste estadístico

Este ajuste α-estable se podría haber programado en un entorno diferente a Matlab. Por ejemplo, en c o en Python, y estos datos haberlos exportado a Matlab para su estudio y representación. Las razones por la que se ha elegido hacer todo el proceso en Matlab han sido:

- El código del TFG de partida había sido programado con Matlab.
- Tener los parámetros α-estables guardados en archivos ".mat", formato propio del Matlab.
- Teniendo todos los datos en un mismo entorno no hay problemas de compatibilidad.
- La librería stbl-master que se puede encontrar en GitHub, de Matlab, facilita el cálculo de los parámetros α-estables.
- Matlab permite representar de forma gráfica los resultados obtenidos en cada apartado para su tratamiento visual.
- Permite operar con filas y columnas con mayor facilidad, de hecho, en la propia clasificación se ha operado con columnas.

3.4 Construcción del espacio de fases

Como se ha explicado en la sección anterior, una vez se tienen calculados los parámetros α-estable para la zona del ataque se procede a la construcción del espacio de fases. Para cada segundo se obtienen los cuatro parámetros en base a los valores contenidos en la ventana de 5 minutos. En la siguiente figura se muestra un ejemplo de un espacio de fases formado por (α, β).

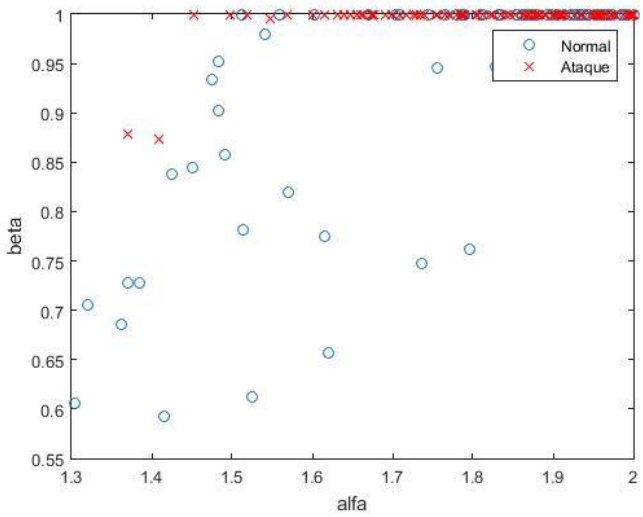


Figura 3-4: alfa vs beta

La metodología consiste en seleccionar pares de parámetros, por ejemplo, α frente a β, donde tiene sentido hacer esta clasificación distinguiendo entre los dos tipos de tráfico. Esta representación constituye un espacio de fases, en la que sólo en algunas combinaciones se puede apreciar que hay cierta separación espacial entre el tráfico mezclado con ataque y tráfico real sin él. Esas diferencias son las que se van a aprovechar para intentar hacer una segmentación de regiones dentro de los planos que forman estos parámetros. De hecho, las figuras de la sección previa son ejemplos de espacios de fases, pero con diferente tamaño de ventana.

3.5 Segmentación

Una vez que se tenían construidos los espacios de fases, para llevar a cabo el estudio se estudiaron diferentes algoritmos y funciones de Matlab que pudieran llevar a cabo la clasificación.

Una vez decididos cuáles eran los pares idóneos para llevar a cabo nuestro estudio se procedió a utilizar diferentes algoritmos empezando por el k-means. Debido a que el tráfico solo estaba separado en 2 clases, el algoritmo k-means al ser aprendizaje no supervisado resultaba que no hacía bien esta clasificación. El problema que tienen los algoritmos de aprendizaje no supervisado es que si la clasificación no está clara no funcionan bien. En la siguiente figura se ve la prueba de ello. Las aspas negras representan

la posición final de los dos centroides después de que el Matlab llevara a cabo la clusterización.

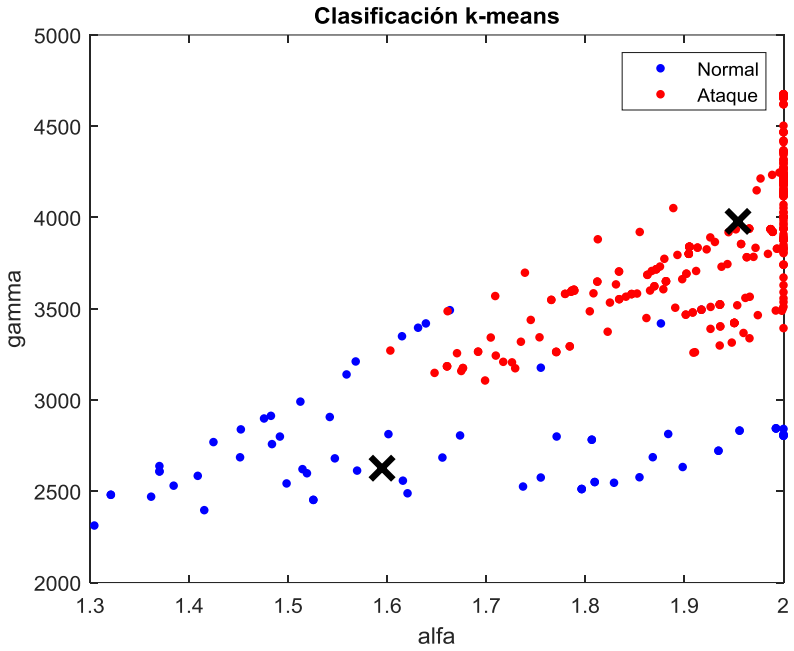


Figura 3-5: K-means para alfa-gamma

Viendo el deficiente resultado de k-means se decidió optar únicamente por algoritmos de otro tipo: aprendizaje supervisado. Esto permitía hacer la clasificación uno mismo previamente, y utilizar algoritmos de aprendizaje supervisado como el análisis discriminante para llevar a cabo nuestro estudio. Al final, k-means trata los datos sin ningún tipo de consideración previa, que para el caso de estudio no es un algoritmo viable, aunque en otras muchas aplicaciones funcione muy bien. En el siguiente capítulo se explicará como se llevó a cabo la clasificación con los algoritmos de aprendizaje supervisado y cómo se hizo la clasificación de los parámetros.

3.6 Metodología

El proceso general de todo el trabajo se va a explicar aquí. En la etapa de desarrollo se expondrá cómo se ha ido haciendo todo este proceso. Las etapas que ha tenido este proceso son las siguientes:

- Una vez que se tenían los parámetros α-estables calculados lo primero que se hizo fue calcular las derivadas de estos parámetros. Al tratarse de datos discretos se calculó la primera y segunda diferencia de estas series de datos, que eran la velocidad y la aceleración. Se analizaron estas derivadas y se descartó una posible clasificación.

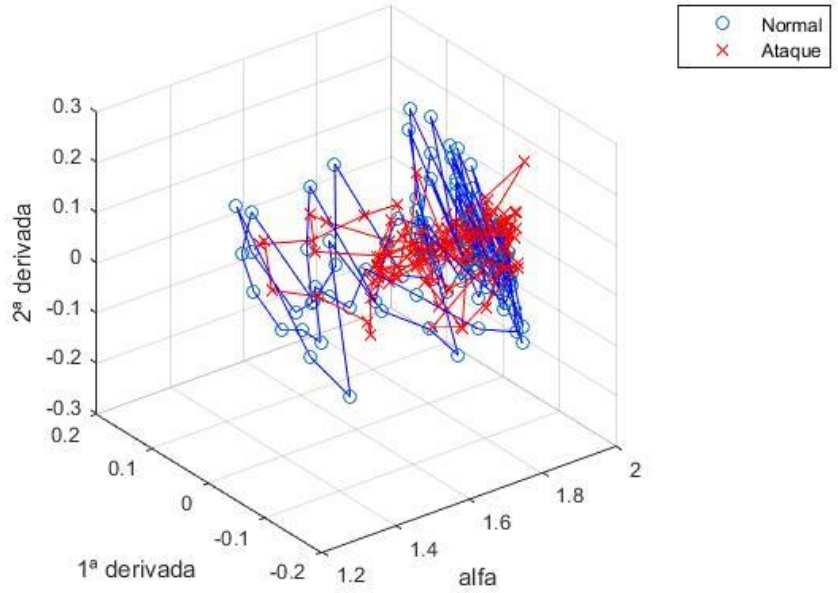


Figura 3-6: alfa vs alfa' vs alfa''

- Con la posibilidad de las derivadas descartada, se construyeron los espacios de fases formados por los parámetros α-estables en combinaciones de 2 en 2. En total había 6 representaciones diferentes. También hay que aclarar que teóricamente son 12 combinaciones, pero lo único que diferían era el eje en el que estaba cada parámetro. Tras un análisis que se explica en la etapa de desarrollo se llegó a la conclusión por inspección de que los pares (α, δ) y (γ, δ) eran los que mejor se podían clasificar y sobre los que se ha hecho el estudio. Además, también se concluyó que el parámetro más significativo de los cuatro era δ.

- Por último, cuando se obtuvo la clasificación de las 2 combinaciones, se hizo un estudio juntando las dos clasificaciones. Se llegó a la conclusión de que juntando estas dos clasificaciones aumentaba ligeramente la tasa de acierto al detectar un ataque.

3.7 Conclusiones

A lo largo de este capítulo se han explicado todas las consideraciones previas que se han tenido en cuenta para realizar el análisis completo de los datos proporcionados. En el siguiente capítulo se explicará con detalle todos los procedimientos y el desarrollo que se ha realizado para lograr el objetivo final que es la segmentación y clasificación de estos parámetros.

4 Desarrollo

4.1 Introducción

En esta sección se explican todos los procedimientos llevados a cabo, incluyendo funciones de Matlab, en base a las decisiones tomadas en la etapa de diseño. Los pasos del desarrollo son:

- Análisis de trafico
- Parámetros estadísticos significativos
- Aplicación de algoritmos de clasificación

4.2 Análisis de tráfico

A lo largo de todo este TFG únicamente se ha usado Matlab. Esto es debido a que todos los datos necesarios para hacer el cálculo de los parámetros α-estables ya venían en formato ".txt" obtenido del trabajo anterior mediante la herramienta AWK que facilita en procesamiento de archivos de texto línea a línea.

Asimismo, se ha aprovechado todo el código proporcionado por el autor del TFG de partida para hacer el cálculo de los parámetros α-estables ya que este estaba en Matlab por lo que todos los cálculos se han realizado con funciones y código de este programa. Para el cálculo de los parámetros hubo que descargarse una librería pública: stlb-master. Esta librería contiene todas las funciones necesarias para el cálculo de los α-estables.

A partir de ahí se ha ido desarrollando todo el código haciendo uso de las posibilidades del Matlab para hacer las representaciones, analizarlas, la posterior aplicación de algoritmos, etc. Dos de las funciones más importantes que se explicarán en los puntos siguientes son classify() y fitcdiscr(). Con estas dos funciones hemos podido hacer la clasificación y segmentación de los parámetros elegidos. Se puede dividir en proceso en 2 partes: cálculo y representación de parámetros, aplicación de algoritmos de clasificación y análisis. En la primera parte se hizo lo siguiente:

- Una vez estudiado y analizado el código y los datos suministrados se procedió a la obtención de los parámetros α-estables para las tres ventanas propuestas, para el tráfico normal como para el tráfico de ataque. Había que tener especial atención en la separación del punto inicial y final de cada ventana para que sólo se tuviera en cuenta la zona de ataque. Esto se hizo para bits como para paquetes.

- Se comenzó por la obtención de los parámetros con los datos en bits, ya que en el TFG anterior se llegó a la conclusión de que los parámetros obtenidos en relación a los bits eran prácticamente idénticos entre el tráfico normal y de ataque. Esto se debe a que el tamaño de los paquetes DoS es muy pequeño, ya que la intención es colapsar el servidor a base de paquetes SYN u otros tipos, pero siendo el tamaño de estos muy pequeño. En la siguiente gráfica se puede ver claramente este efecto:

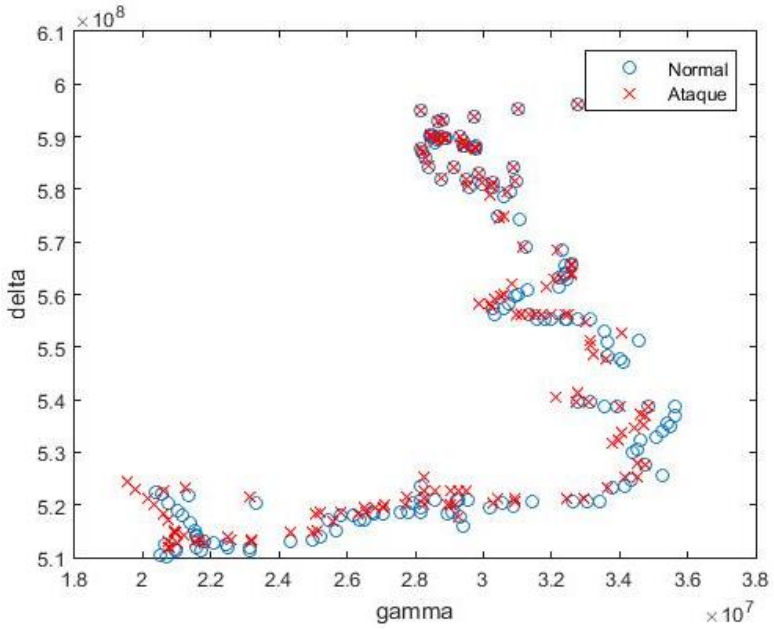


Figura 4-1: gamma-delta en bits

- Una vez descartados los datos en bits, se calcularon y representaron para los datos en paquetes, todas las posibles combinaciones de estos parámetros de 2 en 2. Tras analizar todas estas representaciones se llegó a la conclusión de que la ventana que servía para hacer la clasificación era la de 5 minutos. Alternativamente también se calcularon las derivadas primera y segunda para cada parámetro y también fueron representadas. En las siguientes gráficas se representan la primera y la segunda derivada del parámetro γ, y se observa la dificultad de hacer una clasificación por lo que al final se descartó esta aproximación.

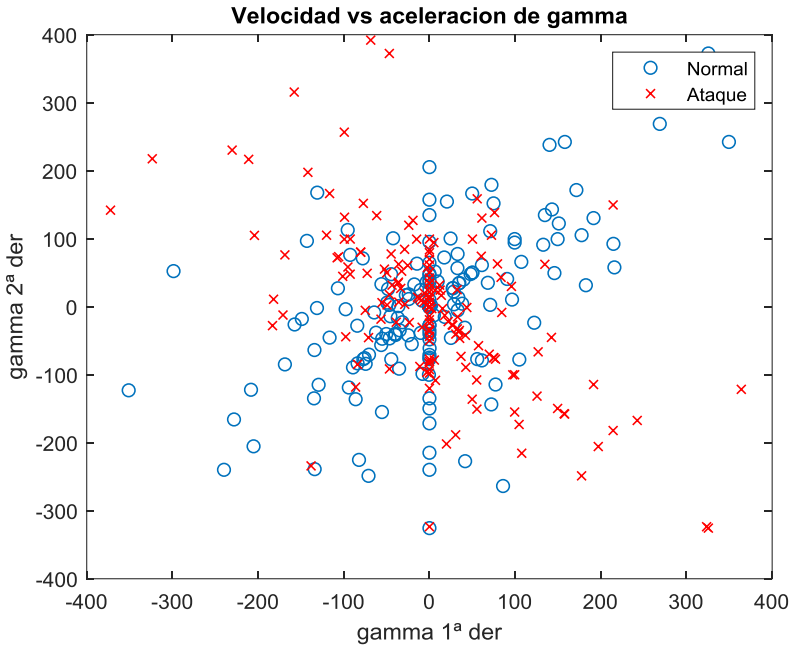


Figura 4-2: Primera derivada gamma

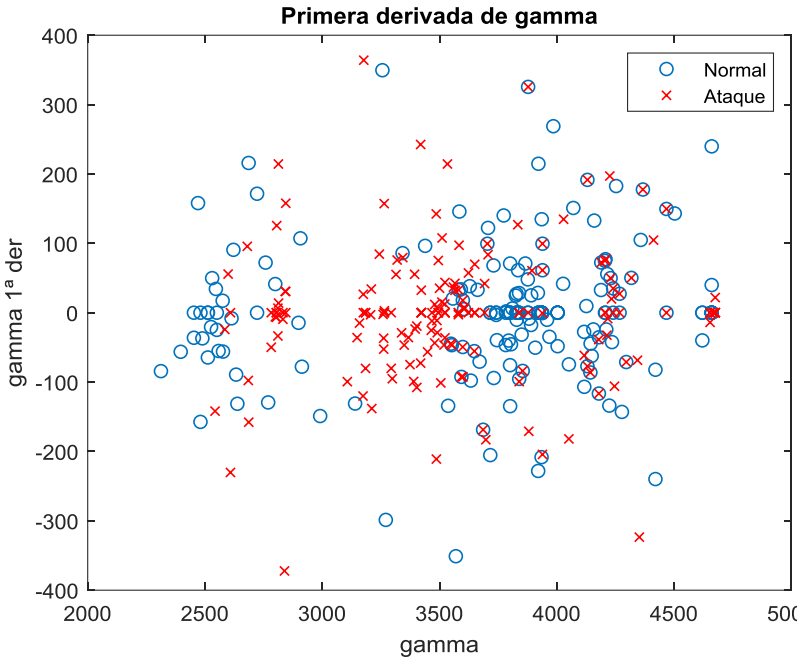


Figura 4-3: Velocidad vs aceleración

- Seguidamente, se estudió la posibilidad de hacer representaciones en 3D de los propios parámetros y estudiar su comportamiento por si resultara útil a la hora de evaluar cuales eran los parámetros que se van a clasificar. En la siguiente figura se representan los parámetros α-β-δ. Se muestran también las líneas que unen puntos sucesivos (correspondientes a ventanas también sucesivas).

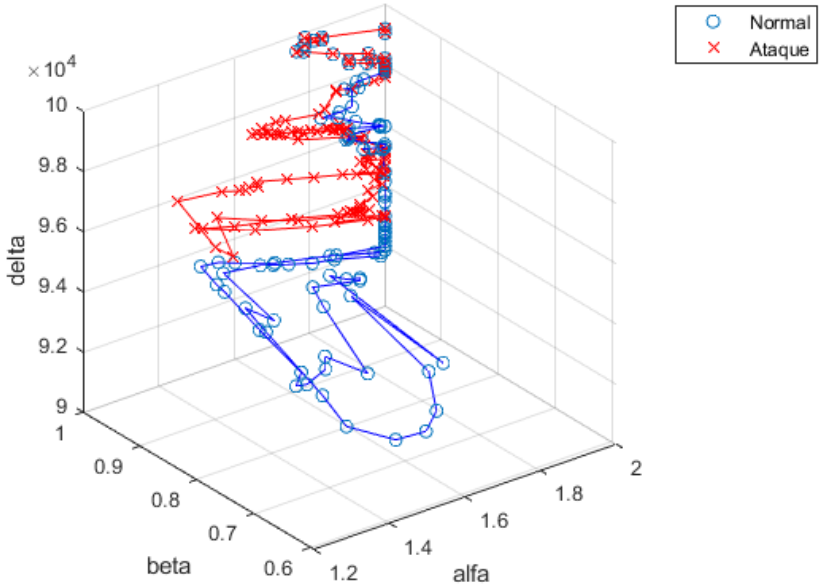


Figura 4-4: alfa-beta-delta

- Posteriormente, mediante un análisis más exhaustivo que se detallará en el siguiente punto se llegó a la conclusión de que las representaciones que era más viables de clasificar fueron los pares (α, δ) y (γ, δ).

- Para facilitar el análisis se han desarrollado funciones parametrizadas que estandaricen todas las representaciones de tal manera que se pueden utilizar para cualquier tamaño de ventana y cualquier parámetro estadístico (α, β, γ, δ).

La segunda parte consistió en aplicar diferentes algoritmos de clasificación como el análisis discriminante lineal, cuadrático y el método Mahalanobis. En esta parte se realizó lo siguiente:

- Como se ha explicado anteriormente, se decidió que la mejor idea para clasificar los parámetros era mediante el aprendizaje supervisado, en el que la clasificación ya está hecha. Primero se usó la herramienta *Classification Learner* de Matlab, que es una herramienta con una interfaz visual que facilita el análisis. Para testear este programa, se le pasaron los datos y la clasificación (todo en una única matriz) y se seleccionó el algoritmo de árbol de decisión. En la siguiente figura se muestra el resultado de la clasificación:

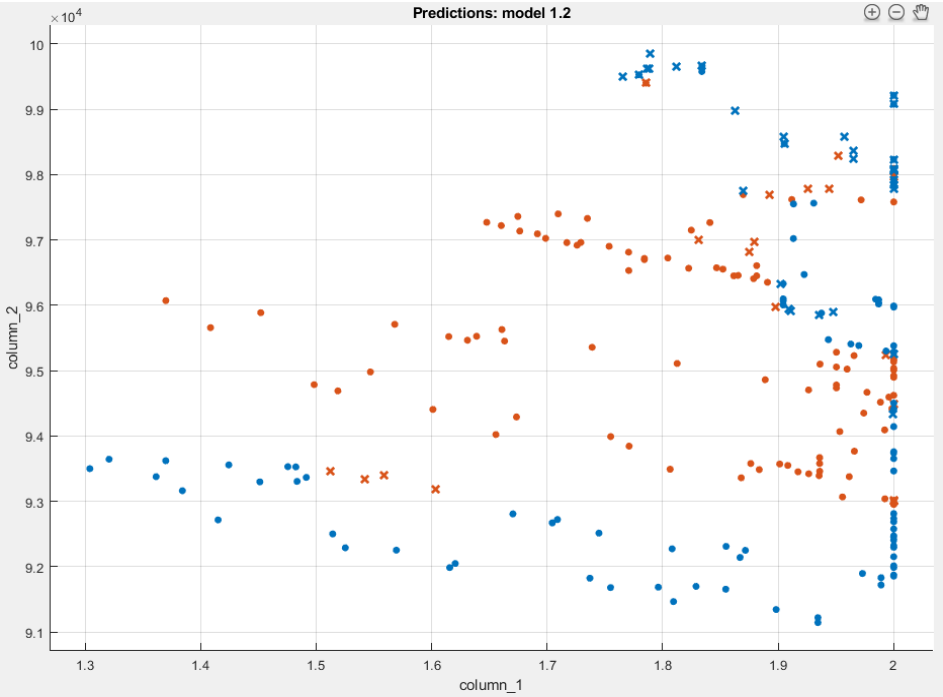


Figura 4-5: Clasificación Medium-Tree

Este modelo dio una precisión del 75.4%, que es moderadamente alta. El problema de esta herramienta es que no se pueden obtener fácilmente las ecuaciones de las rectas que usa, además de que la representación es bastante tosca ya que no se puede cambiar el nombre a los ejes. Por tanto, en lo que se refiere a este trabajo, la herramienta se descartó.

- Por tanto, la solución final adoptada fue escribir un script directamente y llamar a las funciones mencionadas antes: classify() y fitcdiscr(). En la sección 3.5 se explica todo lo relacionado con estas funciones y algoritmos.

4.3 Parámetros estadísticos significativos

Como se ha dicho antes, en este punto se explicará como ha sido el análisis de todos los parámetros para haber llegado a la conclusión de que los pares (α, δ) y (γ, δ) fueran los mejores posicionados para hacer la clasificación.

Echando un vistazo a las figuras, se observó que un porcentaje muy alto de valores de β era igual a 1. Este hecho se puede ver muy bien si se representa la velocidad (1ª derivada) frente a la aceleración (2ª derivada), ya que la mayoría de los valores son cero. Además, se calculó cuál era el porcentaje de valores de β=1, y dio un resultado del 92%. Con todo esto se llegó a la conclusión de que cualquier combinación en la que estuviese β no parecía útil para la clasificación.

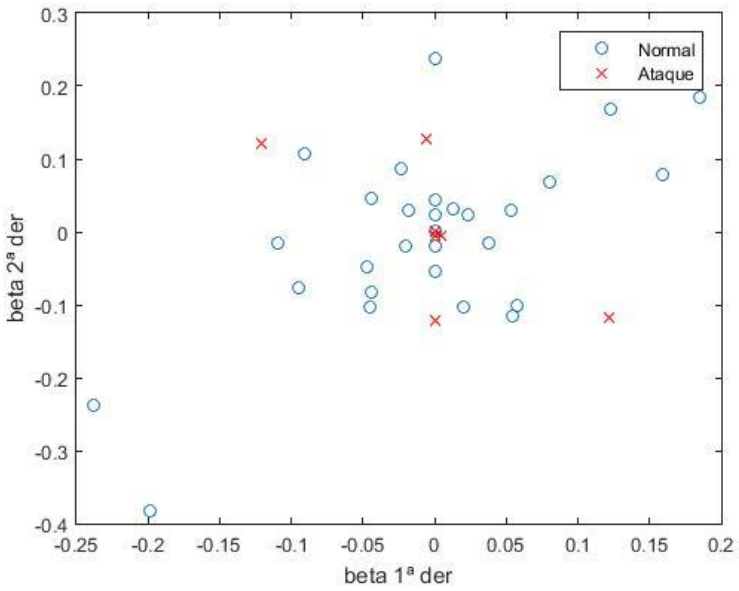


Figura 4-6: Velocidad vs aceleración de beta

En este punto, de momento hay tres parámetros potenciales de ser clasificados. Ahora se va a explicar el proceso seguido para descartar el par (γ, δ). En la siguiente figura se muestra la representación de α frente a γ. Observando la gráfica se puede llegar a la conclusión de que tanto el tráfico de ataque como el normal comparten el mismo espacio de fases, por lo que esta combinación quedó descartada desde el principio al no haber una separación de espacios entre ambos tipos de tráfico.

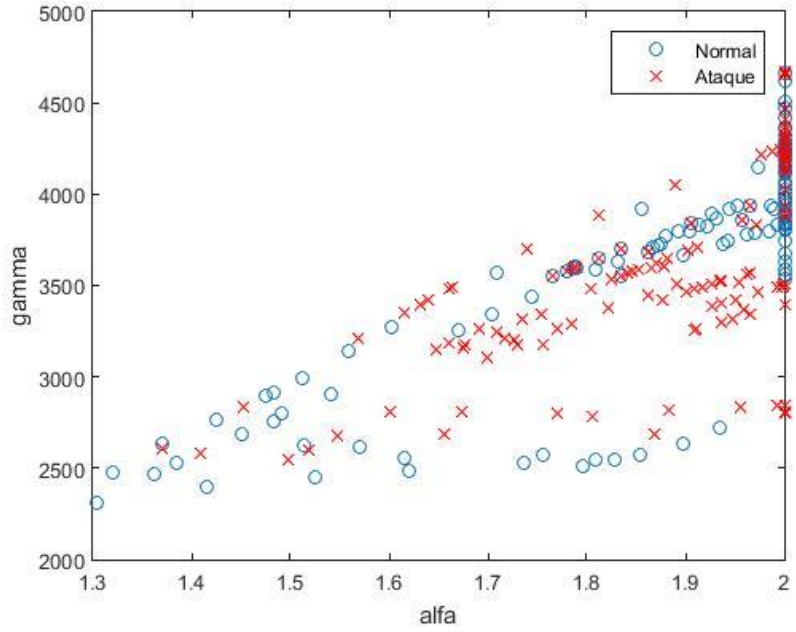


Figura 4-7: alfa-gamma 5 min

Como β ya se había descartado antes, la única opción restante era el par (γ, δ). Observando esta gráfica se puede hacer un descubrimiento interesante: se podría decir que hay un desplazamiento de los valores de γ del tráfico de ataque hacia la izquierda. Este desplazamiento se hace más claro y notorio en los valores de δ comprendidos entre $9.4x10^4$ y $9.6x10^4$.

Este desplazamiento se puede traducir en una posible segmentación de estos parámetros mayoritariamente en esa zona, por lo que también se consideró esta combinación de parámetros a la hora de hacer uso de los algoritmos de clasificación. Esto no quiere decir que se haga una buena clasificación, sino que simplemente se estaría hablando de una posibilidad.

Como conclusión de esta sección podríamos decir que el parámetro más significativo de todos es δ, ya que se tiene en cuenta para las dos combinaciones que se van a estudiar. Además, también se puede afirmar que β es el parámetro menos significativo por su elevado porcentaje de valores iguales a 1 a lo largo de toda la serie.

De hecho, en la siguiente sección de explica cómo se hizo la clasificación, en la que únicamente se ha tenido en cuenta los valores que tomaba δ, independientemente del resto.

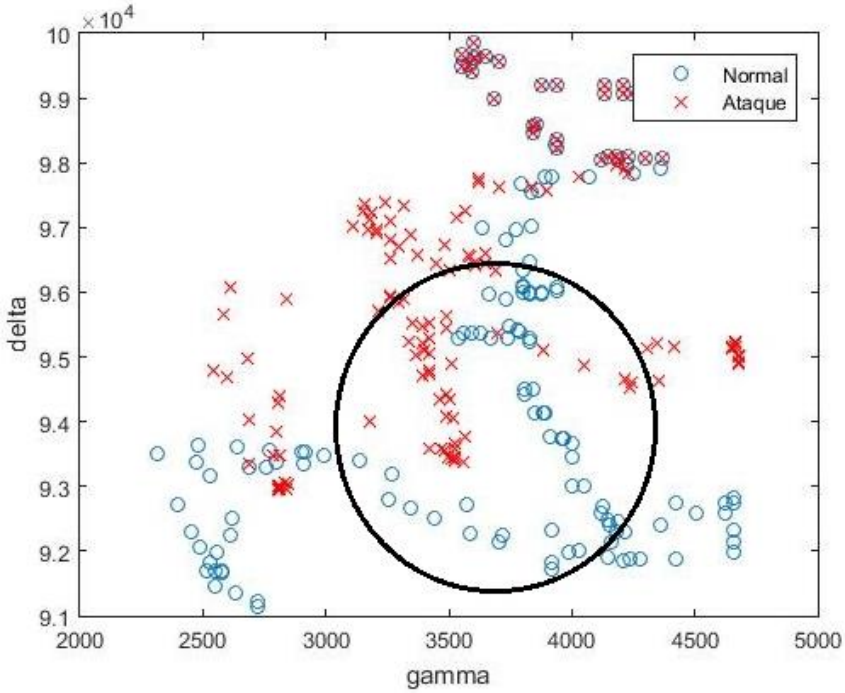


Figura 4-8: gamma-delta 5 min

4.4 Aplicación de algoritmos de clasificación

Tras haber decidido cuales son los parámetros más significativos, que realmente son los que nos dan información de si se está produciendo un ataque o no, se descartó k-means y los algoritmos de aprendizaje no supervisado en general porque los resultados no eran los esperados.

Como siguiente paso, se comenzó a investigar cuáles eran los algoritmos más eficientes para los tipos de datos que se manejaban. Debido a las limitaciones de la herramienta *Classification Learner* se decidió que la mejor idea era programarlo directamente en un script.

Para empezar, se cargaron los datos en una única matriz. Aprovechando que esto ya se había hecho para utilizar la herramienta *Classification Learner*, lo único adicional que hubo que hacer fue añadir una tercera columna referida a la clasificación de los datos. Una vez clasificados, se hizo uso de la función *classify()* a la que había que pasar como argumentos:

- El espacio que ocupaban las dos variables dividido en regiones. El parámetro α estaba comprendido entre 1.3 y 2. $\alpha \in [1.3, 2]$. La variable δ estaba comprendida entre $9.1x10^4$ y 10^5. $\delta \in [9.1x10^4, 10^5]$. Para que la cuadrícula no fuera excesivamente grande y el ordenador pudiera hacer todos los cálculos el espacio de α se dividió en regiones de 0.01 unidades y el de δ se dividió en regiones de 100 unidades. En total la cuadrícula tenía 6400 puntos en total. Se siguió el mismo procedimiento para γ.

- Los parámetros que se querían clasificar, en columnas. En este caso las dos columnas que contenían los parámetros de las combinaciones elegidas.

- Otra columna que contenía la clasificación obtenida tras aplicar los resultados obtenidos en la sección 3.4.

- El último argumento sería el tipo de análisis discriminante que se quiere aplicar. A continuación, se muestra un ejemplo se llamada a esta función:

```
[C,err,P,logp,coeff] = classify([x y],data(:,1:2),clase, 'linear');
```

Los argumentos de salida son cinco, de los cuales, los más importantes son los siguientes:

- C: es un espacio del mismo tamaño que el de entrada en el que punto por punto e indica la clase a la que pertenece ese punto.

- Err: es el error de clasificación que produce el algoritmo.

- Coeff: son los coeficientes necesarios para calcular las rectas que generan los algoritmos a la hora de clasificar.

La otra función necesaria para poder hallar las matrices de confusión de cada parámetro, es *fitcdiscr()*. Esta función es diferente a la anterior por el hecho de que no hace falta generar una cuadrícula para hacer la clasificación. Los argumentos de entrada son los siguientes:

- Al igual que en la función *classify()*, los parámetros que se querían clasificar, en columnas, además del clasificador, que es el mismo que se pasa a la anterior función.

- El tipo de análisis, que en nuestro caso es discriminante, y se denota como '*DiscrimType*'.

- El tipo de análisis discriminante, pudiendo ser lineal, cuadrático y todas sus variantes.

El argumento de salida es un modelo donde se detallan todas las variables que se han generado con el análisis. En base a este modelo, se llama a otra función que es la que realmente hace la clasificación, *resubPredict()*. Las siguientes dos líneas muestran un ejemplo de una clasificación discriminante cuadrática, siendo IdaClass las clases predichas.

```
qda = fitcdiscr(data(:,1:2),clase,'DiscrimType','quadratic');
ldaClass = resubPredict(qda);
```

Tal y como estaban clasificados inicialmente los parámetros, el análisis cuadrático discriminante quedó como en la figura siguiente:

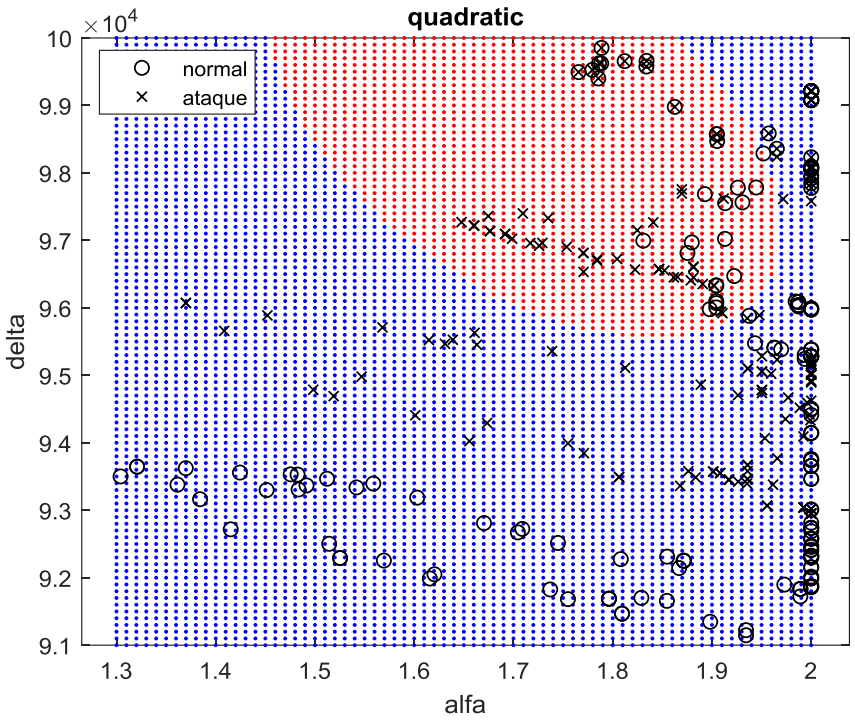


Figura 4-9: alfa-delta quadratic

A la vista está que esta clasificación es muy mala, dando un porcentaje de error del 42%. Toda la zona comprendida entre δ=9.6x10^4 y δ=9.4x10^4 que mayoritariamente es tráfico de ataque no lo clasifica bien.

- Una de las ventajas del aprendizaje supervisado es que uno mismo puede hacer la clasificación más óptima posible, haciendo uso de resultados previos, se puede modificar la clasificación de tal forma que se adapte a los datos que se están manejando. Tanto para el par (α, δ) como para (γ, δ), lo que se hizo fue separar el plano en vez de en 2 regiones (normal y ataque), en 3 (normal1, normal2 y ataque). De esta forma se hizo la siguiente clasificación:

    - Todo lo que estaba clasificado como tráfico de ataque se dejó tal y como estaba.
    - Los valores de δ que estuvieran por debajo de 9.4x10^4 pasarían a ser 'normal1'.
    - El resto de los valores, es decir, lo que estuviera por encima de 9.8x10^4 sería clasificado como 'normal2'.

La siguiente gráfica muestra visualmente y de forma aproximada la clasificación que se hizo:

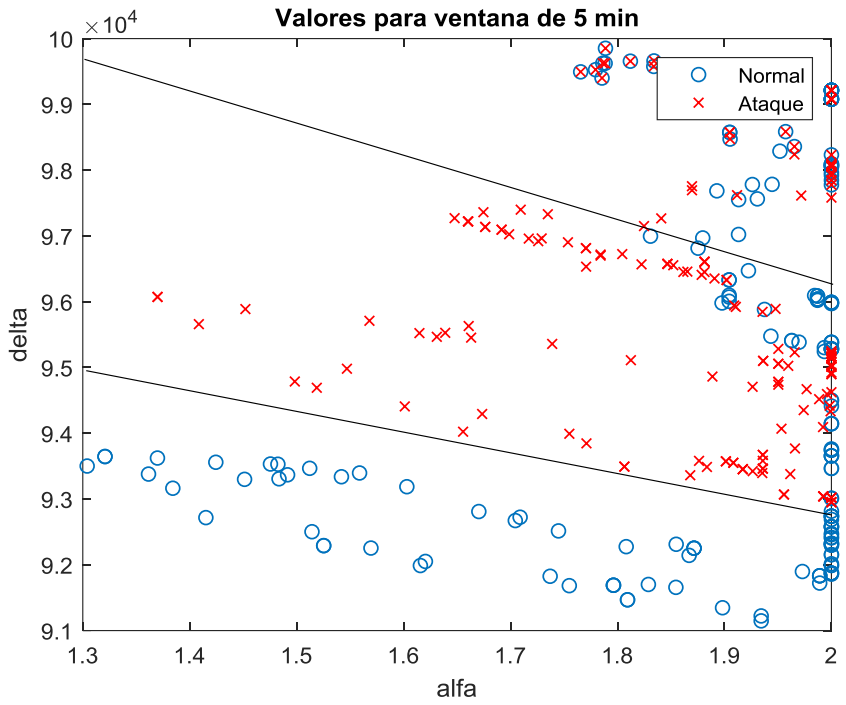


Figura 4-10: alfa-delta regiones

Con esto se consigue una clasificación mucho más precisa donde la zona que interesa que es la del ataque quede segmentada y separada del resto. En la siguiente figura se muestra el resultado de esta nueva clasificación utilizando el análisis discriminante de tipo cuadrático:

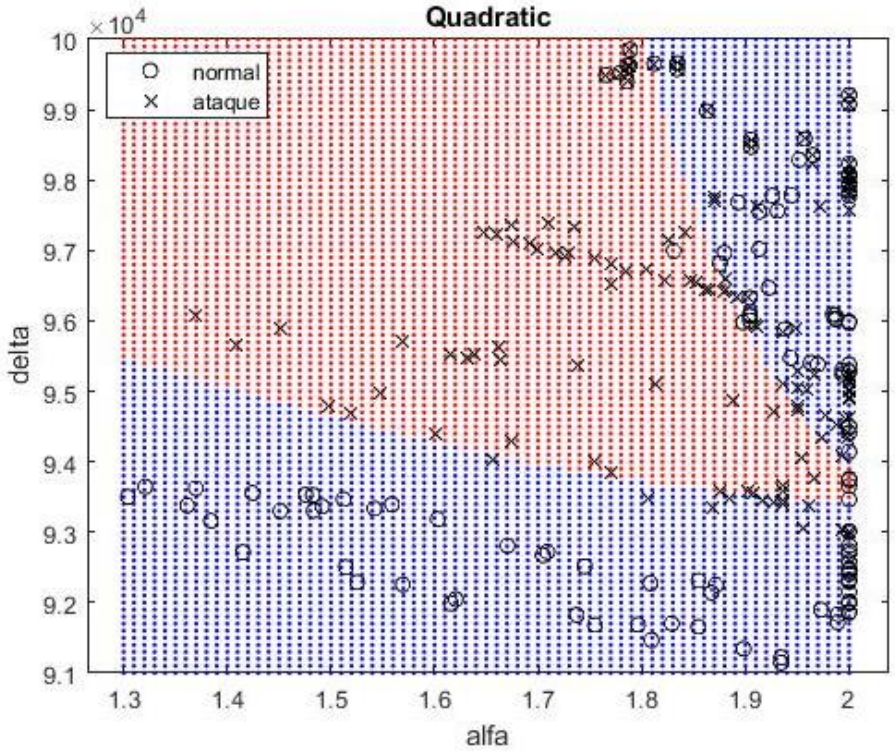


Figura 4-11: alfa-delta quadratic2

En la figura anterior se puede apreciar cómo el algoritmo clasifica mucho mejor, incluyendo dentro del espacio de ataque a la mayoría de los puntos que corresponden al tráfico de ataque. En la sección de pruebas y resultados se explicará con más detalle todos los resultados obtenidos con todos los algoritmos aplicados.

4.5 Conclusiones

A lo largo de toda esta sección se han ido observando y explicando todos los procesos que han sido necesarios para la realización de este trabajo. Gracias a la librería stbl-master se han podido efectuar los cálculos de los parámetros α-estables para posteriormente haber sido clasificados.

Además, se ha explicado todo el proceso de clasificación de estos parámetros, para el que en el siguiente capítulo se detallarán todos los resultados obtenidos.

5 Pruebas y resultados

5.1 Introducción

En este capítulo se presentarán y explicarán todos los resultados obtenidos tras el uso de los algoritmos explicados en el capítulo 2. Se mostrarán todos los tipos de análisis discriminante que se han usado con sus respectivas ecuaciones y tasas de error. En esta sección se hablará de:

- Metodología
- Resultados
- Conclusiones finales

5.2 Metodología

Como se ha ido diciendo a lo largo de todo el trabajo, la única ventana válida es la de 5 minutos, ya que tiene un número de puntos suficiente para realizar el análisis y el cálculo de parámetros tiene menos en cuenta el tráfico normal en la serie temporal mezclada.

Antes de mostrar todos los resultados obtenidos, para ver la importancia que tenía elegir el tramo de análisis se partió de los datos originales y se calcularon todos los parámetros de toda la serie temporal que duraba un total de 4 horas. De este periodo de tiempo se escogió una zona que contuviera no sólo a la zona del ataque si no también a zonas que fueran de tráfico normal. En la primera gráfica se muestra la zona que se tuvo en cuenta para el análisis distinguiendo entre zonas:

- Sólo tráfico normal: para las dos series temporales las zonas de tráfico normal son idénticas por lo que se les asignó solo un color a las dos series. color azul

- Tráfico normal en la zona del ataque: para distinguir claramente las dos series temporales en la zona del ataque, a la serie que sólo contiene en tráfico sin anomalías se le ha designado el color verde.

- Tráfico mezclado: la ventana de 5 minutos contiene tanto a zonas del tráfico normal como a zonas del ataque. Color morado.

- Tráfico de ataque: contiene al tramo completo de 2 minutos que dura el ataque. La duración de esta zona es de 3 minutos ya que, si la ventana es de 5, durante 3 minutos está cogiendo toda la zona de ataque. Color rojo.

También se incluye una gráfica en 3 dimensiones que contiene la representación de las variables α, γ y δ, con la misma distribución de colores.

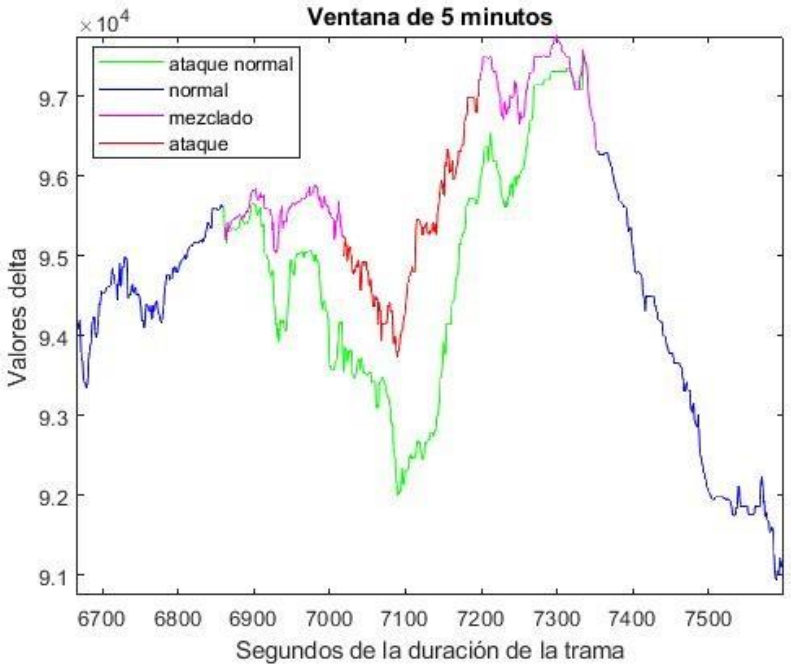


Figura 5-1: parámetro delta por colores

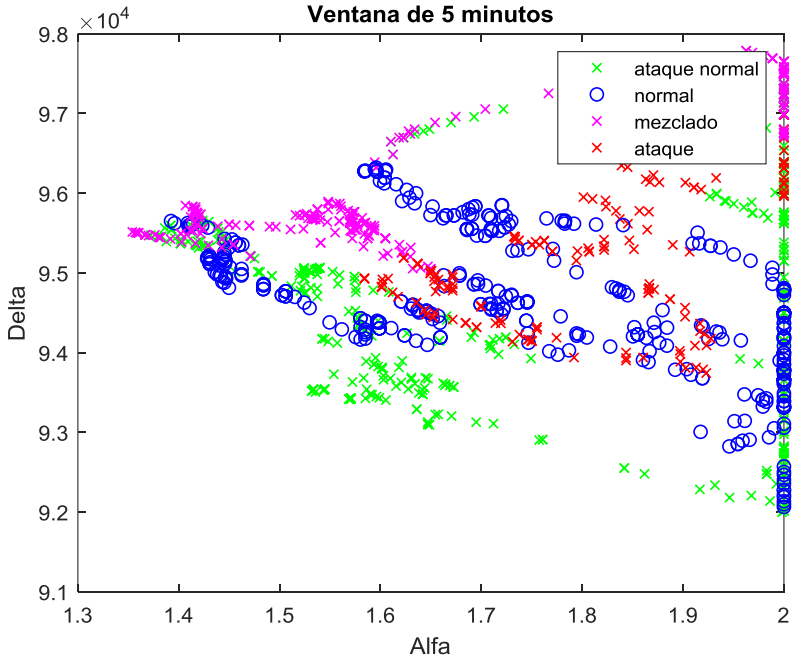


Figura 5-2: alfa-delta por colores

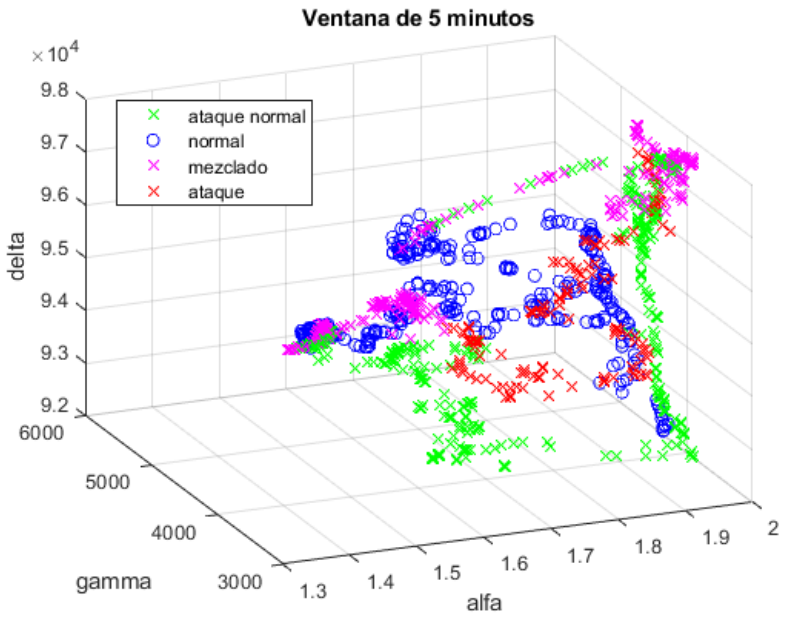


Figura 5-3: alfa-gamma-delta por colores

En cada una de las tres gráficas se hace notoria la importancia de seleccionar bien tanto el punto de inicio como el final del tramo de análisis. El tráfico normal en este caso sólo estorbaría porque no aporta ninguna información útil. Incluso la mayor parte del tráfico mezclado tampoco sería de gran ayuda ya que la mayoría de los valores podrían dar lugar a resultados erróneos en la clasificación. Por tanto, todo eso se eliminó y solamente se tuvo en cuenta la propia zona del ataque.

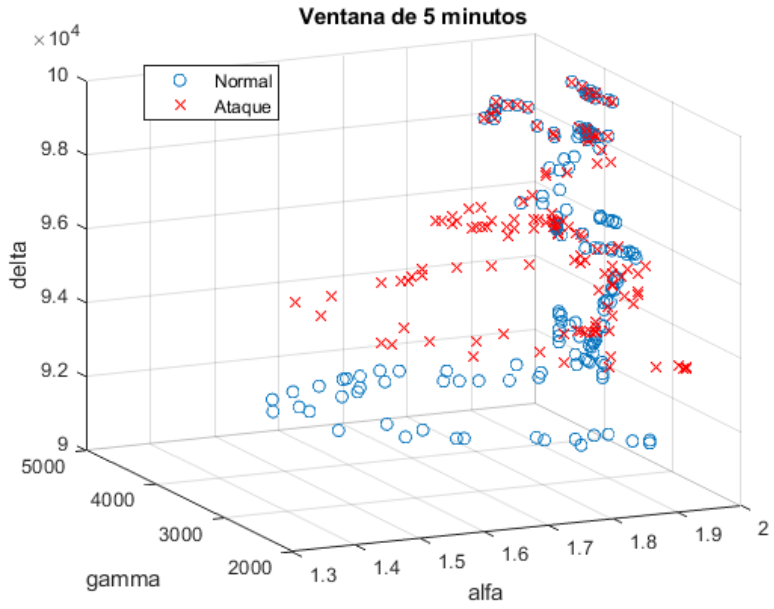


Figura 5-4: alfa-gamma-delta

5.3 Resultados

En base a lo descrito anteriormente se procedió al estudio y entendimiento de todas las funciones de Matlab necesarias para hacer la clasificación. Esta parte ya se ha explicado en la sección 3.3, asique en este punto se comentarán todos los resultados arrojados por estas funciones. Se va a empezar estudiante los resultados del par (α, δ) y posteriormente se hará lo mismo con (γ, δ). Las matrices de confusión, a pesar de que se ha dividido el tráfico normal en 'normal1' y 'normal2', sólo distinguen entre tráfico normal y de ataque, es decir, son binarias.

Primeramente, se va a empezar por el análisis discriminante de tipo lineal. Es el análisis más básico dentro de los discriminantes porque sólo emplea rectas parea hacer la segmentación. Al ser el método más básico es el que peor clasifica. Teóricamente tendrá que ser el método que mejor funcionase ya que las regiones están delimitadas por rectas, pero en la práctica no es así. Se observa como muchos valores de delta son clasificados como tráfico normal cuando no lo son, en especial la zona de $9.5x10^4$. Las rectas que genera este algoritmo son las representadas en color morado como 'data1' y 'data2'.

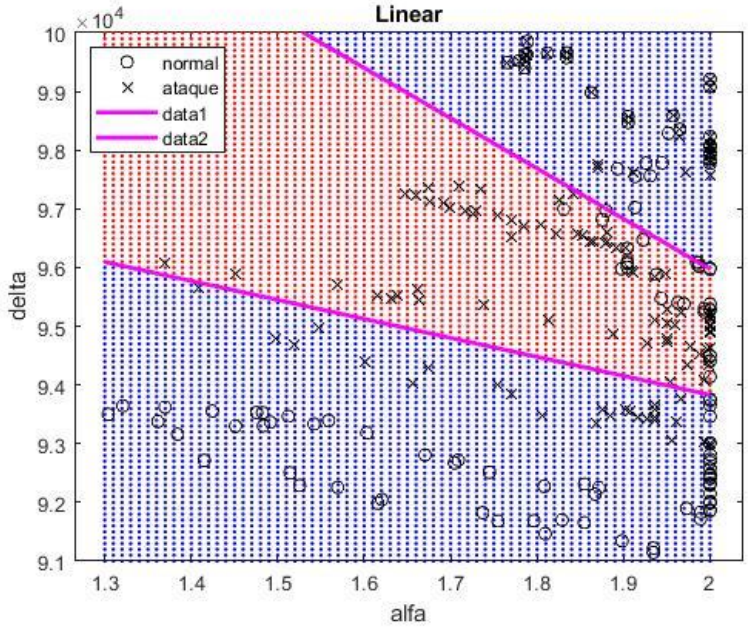


Figura 5-5: clasificación método lineal alfa-delta

Una buena prueba de ello es la matriz de confusión. Es un tipo de matriz con la que se puede ver fácilmente cuáles han sido los aciertos y los fallos en cada clase, pudiendo analizar las zonas de mayor conflicto. En cuanto al tráfico de ataque se puede observar que, de los 181 puntos, clasifica bien 101, con una tasa de acierto del 55.8%, un porcentaje bajo, ya que hay 80 falsos negativos. El tráfico normal se clasifica ligeramente mejor, con 124 de 181, lo que se traduce en un acierto del 68.5%. El acierto global de este método es del 62.15%. Es un porcentaje muy bajo, por lo que esta clasificación no es válida.

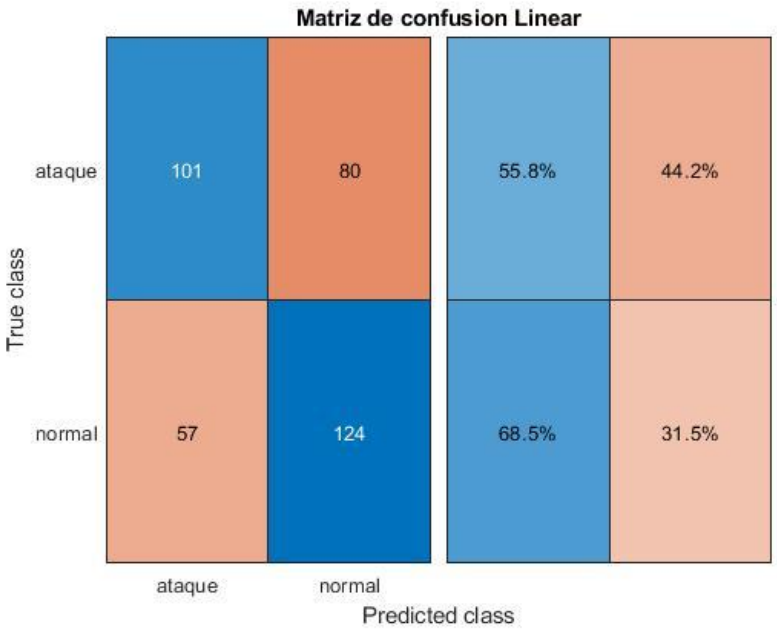


Figura 5-6: matriz de confusión método lineal alfa-delta

El siguiente método es el cuadrático (‘quadratic’). Como se puede observar este método emplea parábolas para hacer la segmentación del espacio. En comparación con la anterior gráfica, aquí si se puede ver mucho mejor una segmentación clara entre la zona de ataque y la zona de tráfico normal. La parábola inferior tiene la forma que dibujan los parámetros en la zona de ataque, aunque algunos los clasifica como ataque. La parábola de arriba responde muy bien a la clasificación previa que se hizo de estos parámetros. Aun así, hay una mezcla en esa zona tanto de tráfico normal como de ataque que no es nada buena para la propia clasificación.

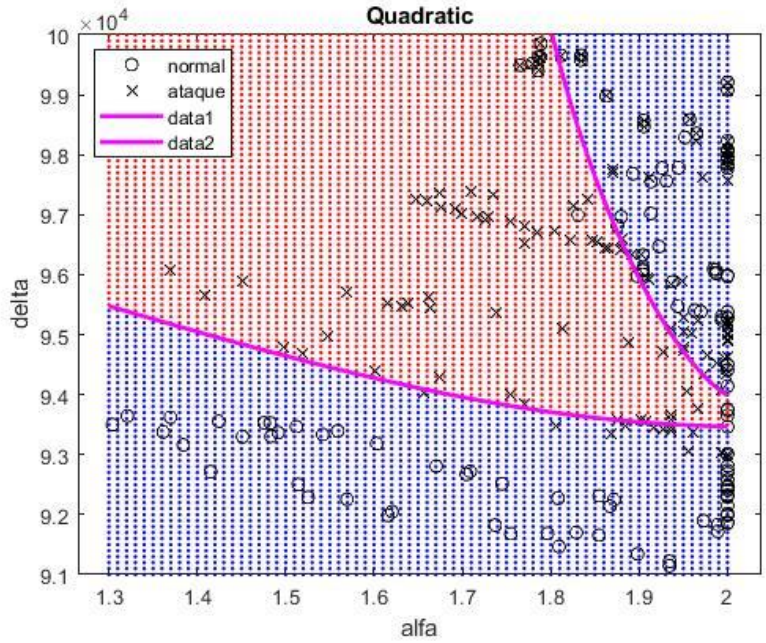


Figura 5-7: clasificación método cuadrático alfa-delta

La matriz de confusión es claramente mucho más favorable que la del método lineal. Hay un notable incremento del acierto de la clasificación tanto del tráfico de ataque como del tráfico normal. En total, hay 127 puntos de ataque clasificados correctamente, lo que significa que el acierto es del 70%, y 144 puntos del tráfico de ataque bien clasificados, que se traduce en casi un 80% de acierto. En consecuencia, hay una reducción importante del número de falsos positivos y falsos negativos decrece. El acierto global asciende hasta el 74.86%. Con estos datos de acierto, la clasificación es aceptable.

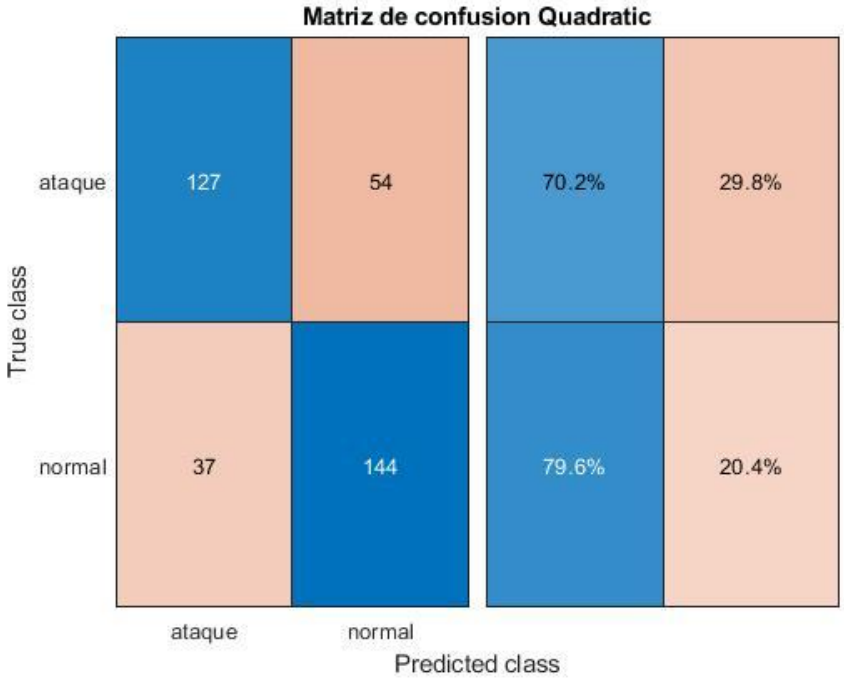


Figura 5-8: matriz de confusión método cuadrático alfa-delta

El último método que se propuso fue el de Mahalanobis, que, como se ha explicado en el capítulo 2, no es más que una distancia entre el punto y la distribución. Para cada punto hay 3 distancias: 'normal1', 'normal2' y 'ataque'. Para elegir a clase de cada punto se escoge la menor distancia de las tres y se le asigna su etiqueta. Es una clasificación parecida a la cuadrática ya que también emplea parábolas para dividir las regiones. La parábola inferior es prácticamente igual que la del método anterior, pero la superior se diferencia más. En el método cuadrático esa parábola empezaba más abajo, aproximadamente a un valor de δ de $9.45x10^4$, mientras que por el método Mahalanobis, la parábola empieza en $\delta=9.6x10^4$. Esto implica que se clasifican más valores de 'ataque' correctamente, a diferencia del método cuadrático.

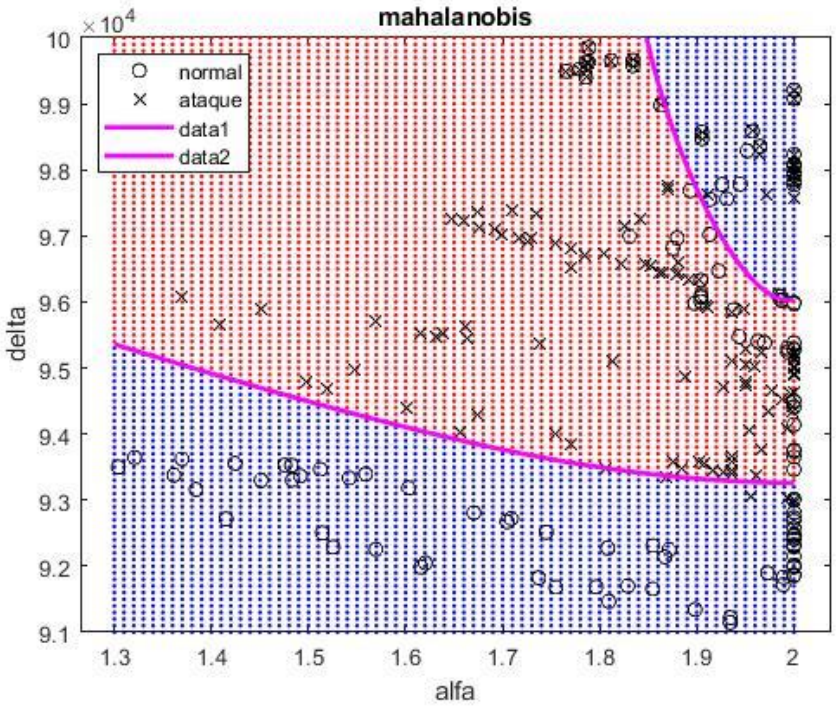


Figura 5-9: clasificación método Mahalanobis alfa-delta

Las matrices de confusión de este método y del cuadrático son parecidas. El tráfico de ataque lo clasifica algo mejor, aumentando de 127 a 132 puntos bien clasificados, y a una tasa de acierto del 73%. No obstante, la diferencia clave entre ambos métodos es clasificación del tráfico normal. El acierto se reduce considerablemente de un 80% que tenía el cuadrático a un 67% del Mahalanobis. Esto hace que la clasificación sea peor, aumentando mucho el número de falsos positivos, de 37 a 60. El acierto total de este método se sitúa en el 70%.

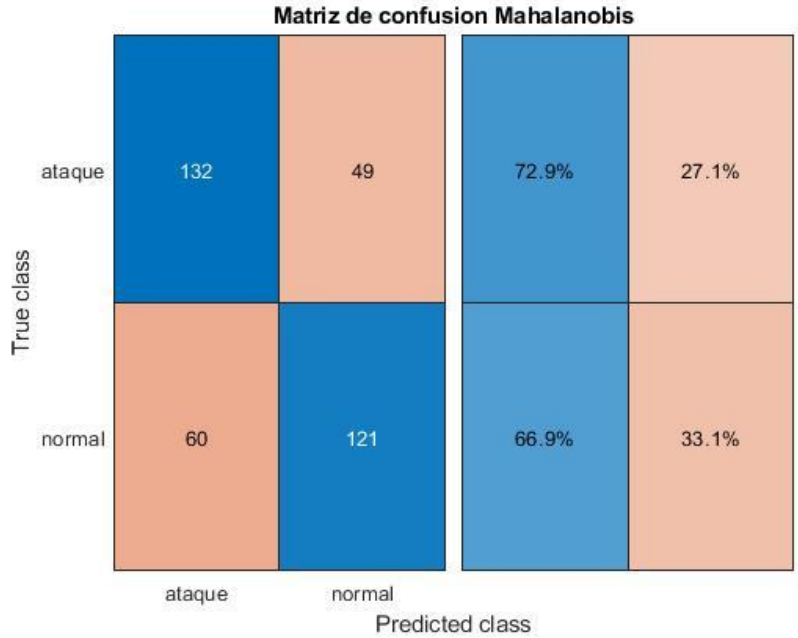


Figura 5-10: matriz de confusión método Mahalanobis alfa-delta

Ahora se van a analizar los resultados del siguiente par, (γ, δ). En el método linear se vuelve a ver una segmentación muy simplista ya que, como se ha dicho antes, se utilizan rectas. Es un tanto curioso que las dos rectas calculadas se crucen a la hora de representarlas. Esta separación se deja muchos puntos de tráfico de ataque como tráfico normal, además, el inconveniente que tiene este espacio de fases es la zona marcada superior donde hay una mezcla de puntos de los dos tipos. Esto no sólo es algo que condiciona a este método, sino a todos los demás.

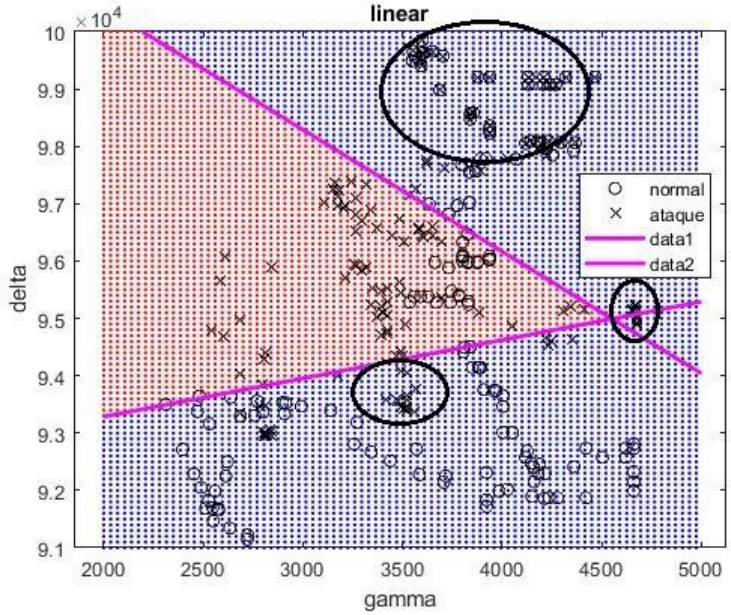


Figura 5-11: clasificación método linear gamma-delta

La matriz de confusión confirma lo que a simple vista se observa. Los porcentajes de acierto de ambos tráficos es muy bajo, ya que ninguno llega al 70%. En cuanto al tráfico normal, el acierto es del 65.2%, habiendo un total de 63 falsos positivos. El tráfico de ataque tiene unos números parecidos, con una tasa de acierto del 60.8%. Al igual que pasaba con (α, δ), el método linear no es válido para el tipo de datos que se clasifican. El acierto total del método linear es del 63%.

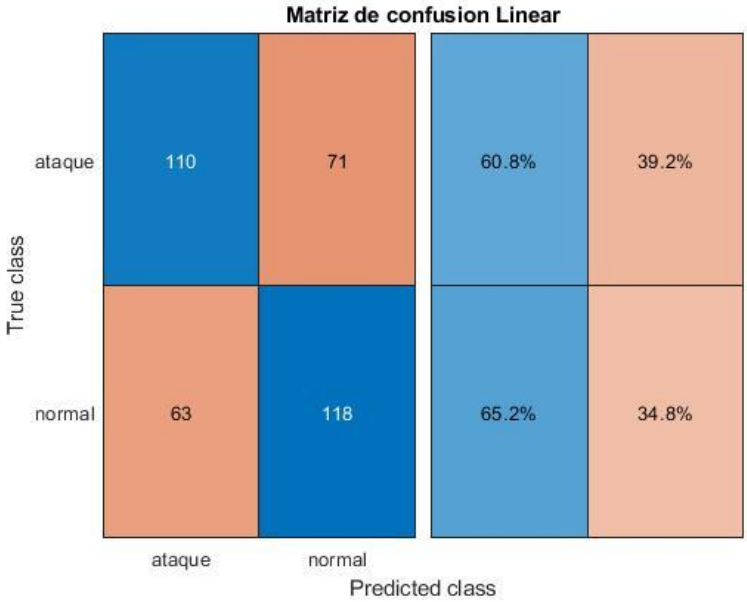


Figura 5-12: matriz de confusión método linear gamma-delta

El método cuadrático es el que mejor clasifica, a simple vista, de los tres métodos expuestos. La mayoría del tráfico de ataque lo clasifica correctamente, pero vuelve a haber conflicto en la zona superior. Como es imposible separar estos valores, independientemente del algoritmo utilizado, en esa zona siempre se va a cometer error.

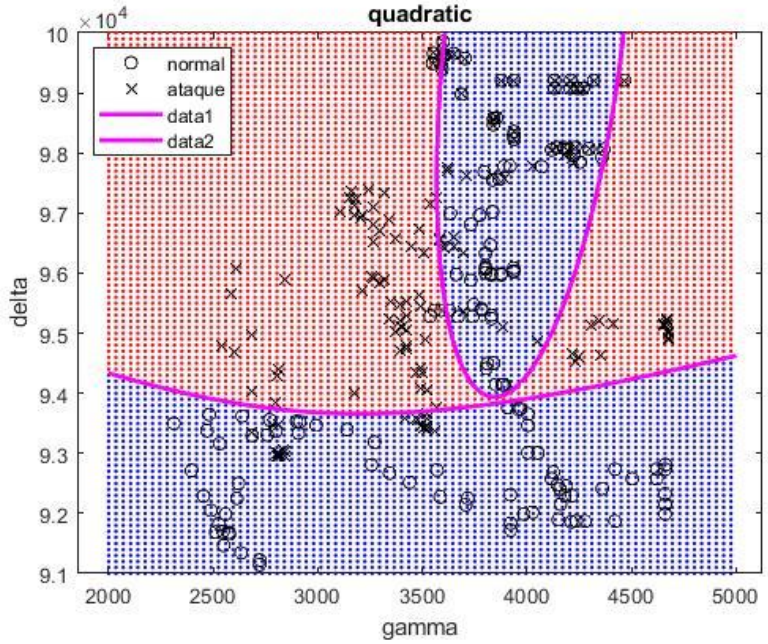


Figura 5-13: clasificación método quadratic gamma-delta

Tomando en cuenta la matriz de confusión, cabe destacar el elevado número de puntos, en concreto 132, que los clasifica correctamente como ataque. Además, hay una reducción considerable del numero de falsos positivos y falsos negativos. En comparación con la clasificación que, hacia este mismo método con el otro par, la diferencia reside a la hora de clasificar el tráfico normal, cuyo acierto es del 74%, siendo inferior al 80% de (α, δ). En conjunto, al acierto total es del 73.5%.

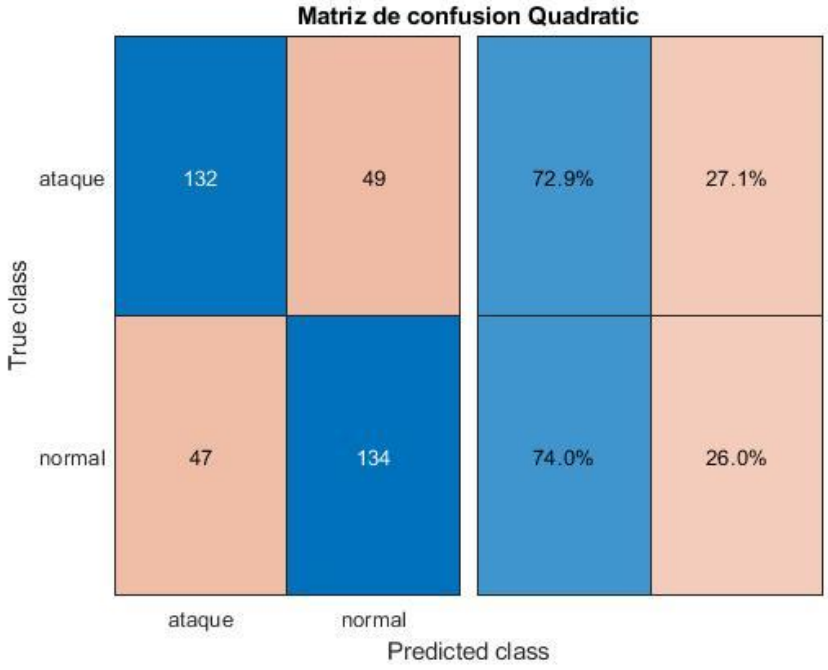


Figura 5-14: matriz de confusión método quadratic gamma-delta

Por último, se llega al método por distancia Mahalanobis. Como ya se ha visto es bastante parecido al cuadrático, aunque con algunas variaciones. En este caso, la zona azul superior es mucho más pequeña que en el cuadrático por lo que se deja más tráfico de ataque como tráfico normal, en la zona marcada con el círculo. Esto va a provocar un aumento significativo del error total, que asciende al 33.15%.

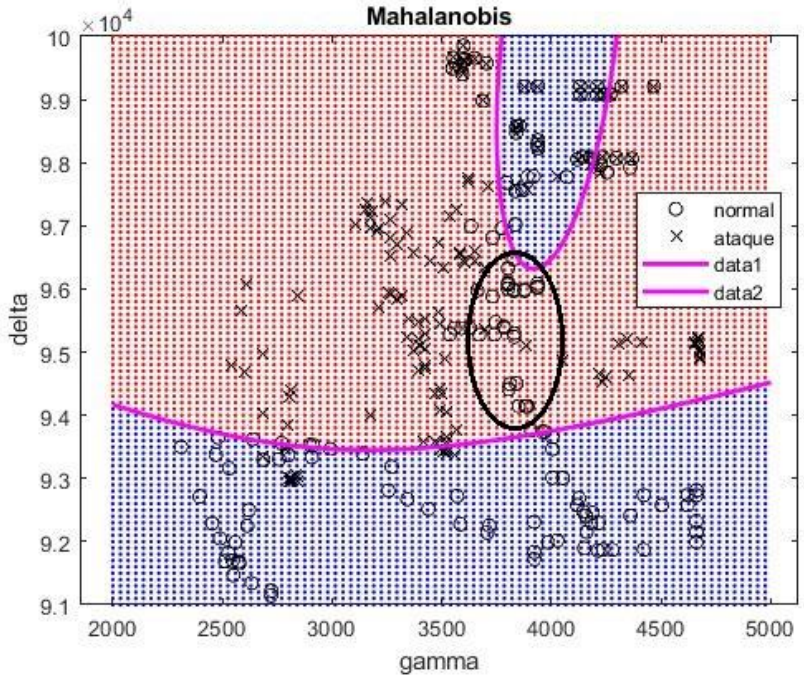


Figura 5-15: clasificación método Mahalanobis gamma-delta

La matriz de confusión revela que este método tiene el mismo acierto clasificando el ataque que el cuadrático, pero como se acaba de explicar, se clasifica mucho peor el tráfico normal. La tasa de acierto desciende del 74% al apenas 61%, incrementando, en consecuencia, el número de falsos positivos. Esto hace que, al igual que con (α, δ), el acierto total sea menor, en concreto, sería del 66.85%.

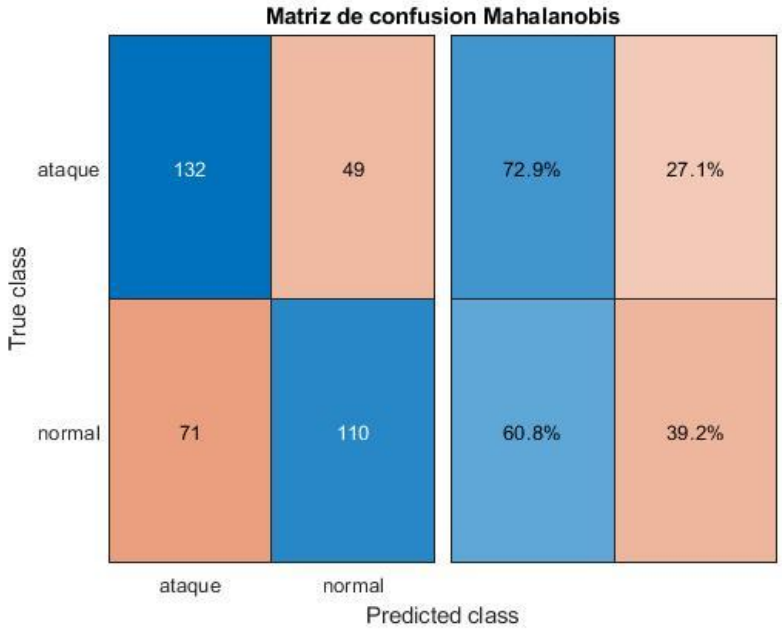


Figura 5-16: matriz de confusión método Mahalanobis gamma-delta

5.4 Conclusiones

En esta sección se van a relacionar todos los conceptos que hemos visto en este apartado para en el siguiente capítulo, acabar de aclararlos y proponer los posibles trabajos futuros.

El mayor problema, y por tanto donde más error se comete, es en las zonas donde coinciden los parámetros tanto para tráfico normal como para el de ataque, esto el algo que por más que se intente separar no se va a poder. Es una zona que por sus características se podría clasificar de una forma diferente a como se ha hecho, tratándola, por ejemplo, como tráfico mezclado, que simplemente no se tenga en cuenta.

Mucho del error que comenten también estos algoritmos es en la zona de $\alpha=2$, en (α, δ). Dado que es un valor que comparten tanto tráfico de ataque como tráfico normal también se hace imposible separarlos. Se estudió la posibilidad de obviar esta región, pero a la hora de representar una variable frente a otra, el Matlab impone la condición de que ambos vectores tengan el mismo tamaño.

Aun así, el porcentaje de acierto es mejor del que se esperaba al principio, aparte de algunos resultados sorprendentes en alguno de los métodos.

En la siguiente tabla se muestra una comparativa más detallada de los aciertos de clasificación que tiene cada método en los dos pares estudiados:

	Lineal	Cuadrático	Mahalanobis
(α, δ)	62.15%	74.86%	69.61%
(γ, δ)	63%	73.49%	66.85%

Figura 5-17: tabla comparativa de errores

En ambos casos los porcentajes son muy parecidos. No obstante, da la sensación de que el método cuadrático aplicado al par (γ, δ) es el que mejor segmenta el espacio. El método linear es el que peor funciona, probablemente porque sea el método más simple, pero no quiere decir que para otro tipo de datos funcione mal, si no que para los datos que se han manejado en este TFG este método no es válido. Estos errores en la mayoría de los casos son inviables para decir que la clasificación es buena, sin embargo, del método cuadrático podría decirse que con un acierto del 75% es un resultado bueno.

El método Mahalanobis visualmente es el que mejor funciona para clasificar (α, δ), pero gracias a otros parámetros y datos como el error cometido y la matriz de confusión, esto no es así. Este hecho resalta la importancia de no fiarse simplemente de una inspección ocular y utilizar procedimientos cuantitativos.

Aunque en el TFG anterior se llegó a la conclusión de que, para una ventana de 15 minutos, el valor α era el que más se diferenciaba en la zona del ataque con respecto al tráfico normal, no se ha encontrado útil para el método de clasificación desarrollado.

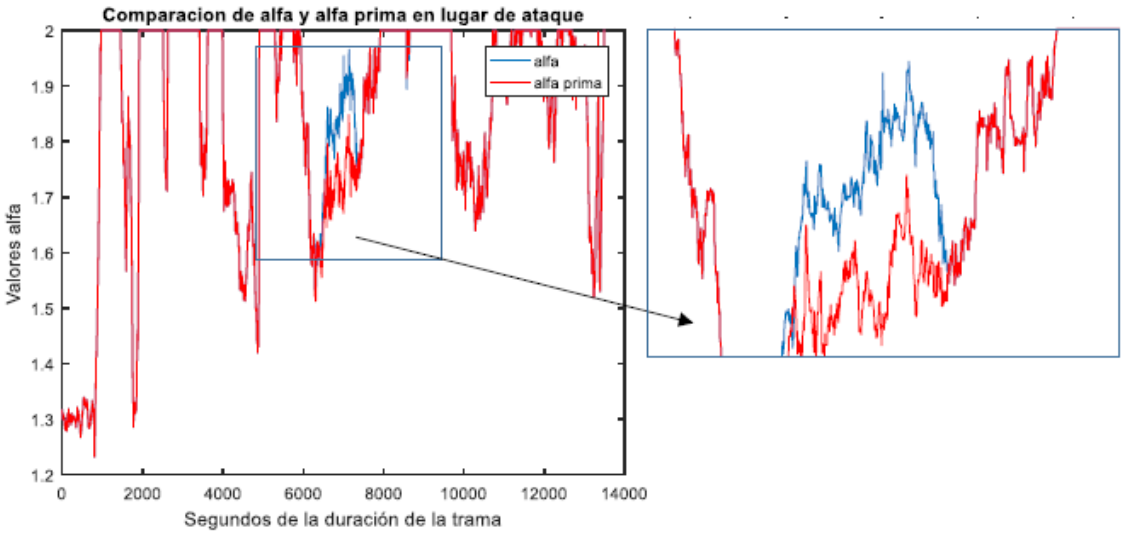


Figura 5-18: representación de alfa a lo largo de 4 horas con ventana de 15 min

Para una ventana de 5 minutos, que es la que se ha elegido para el estudio, el parámetro más representativo ha sido el de δ, ya que la clasificación que se ha hecho ha sido siempre en base a este parámetro. También hay que decir que, al ser ventanas más pequeñas, la diferencia entre las dos series es menor.

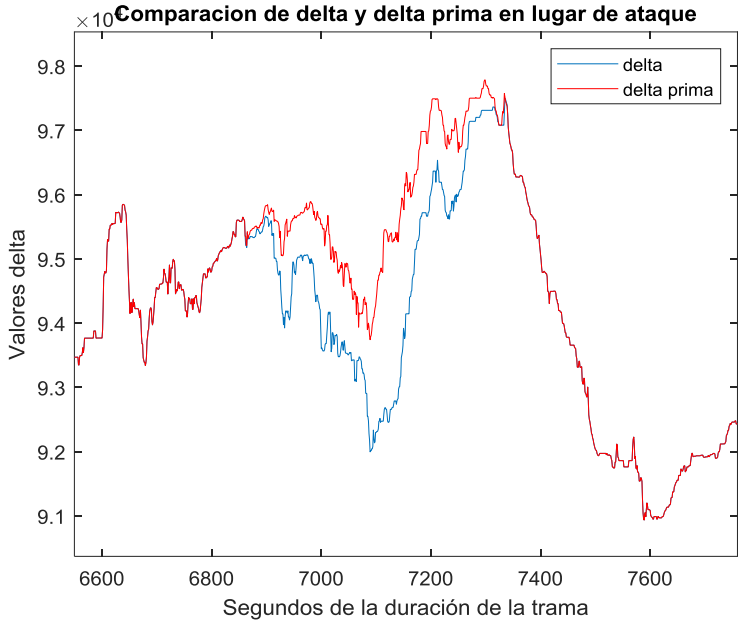


Figura 5-19: representación de delta a lo largo del ataque con ventana de 5 min

Además de todo esto, también se pensó en combinar los resultados de ambos espacios de fases. De esta forma, el hacking que no se detectara en un espacio podría detectarse en otro. Esto es posible debido a que en cada segundo se calculan los cuatro parámetros, por tanto, un punto estaría formado por los valores de cada uno de los cuatro parámetros: (α, β, γ, δ), lo que se traduce en 4 coordenadas. Lo que se ha hecho es aplicar estos algoritmos a dos combinaciones de estos parámetros que eran las más susceptibles para hacer el estudio.

Se estudiaron las dos posibilidades: unión e intersección. Primero se hablará de la unión. La condición de este tipo de probabilidad es la detección de ataque en un espacio o en otro, es decir, si se detecta ataque en uno de los dos espacios, el tráfico se clasifica como 'ataque'. Como se puede observar, el acierto es ligeramente mayor, llegando hasta el 76%, pero es una mejora muy leve. En consecuencia, el número de falsos positivos aumenta notablemente de una media de 42 entre los dos casos a un total de 60, disminuyendo también el acierto de clasificación del tráfico de ataque. Esto provoca el aumento de falsas alarmas de ataque, pero es mejor una falsa alarma que no detectar un hacking.

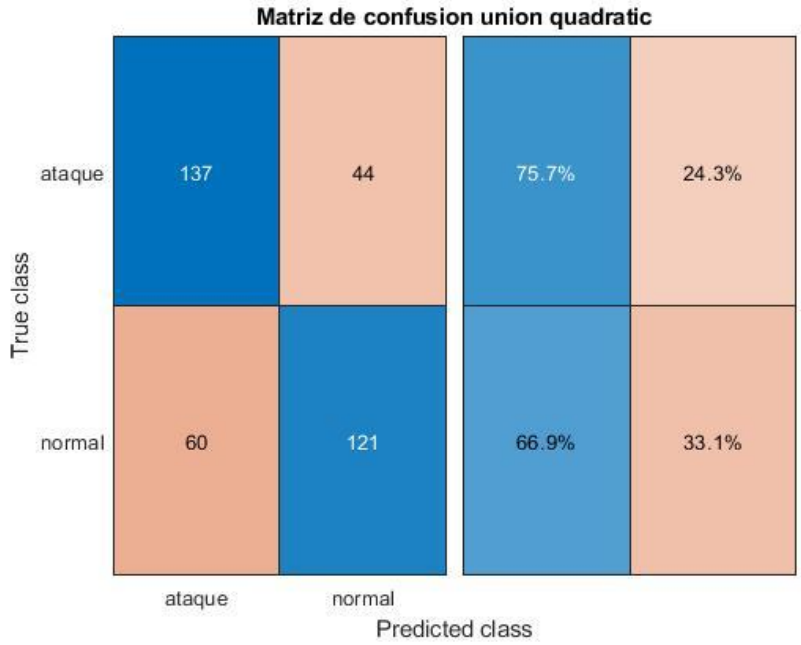


Figura 5-20: matriz de confusión de unión quadratic

Seguidamente se estudió el segundo caso, la probabilidad conjunta o intersección, en la que sólo se clasificaba como ataque si en ambos espacios de fases era clasificado como ataque. De esta forma, el porcentaje de acierto de la clasificación del tráfico de ataque descendía al 67.4%, mientras que el acierto de clasificación del tráfico normal mejoraba notablemente hasta el 87%. Esto se debe a la propia condición de la probabilidad, ya que, si se detectaba ataque en un espacio mientras que en el otro no, el tráfico se consideraba como normal. Asimismo, el número de falsos negativos aumenta hasta 59, mientras que se reduce el de falsos positivos, con tan solo 24 puntos.

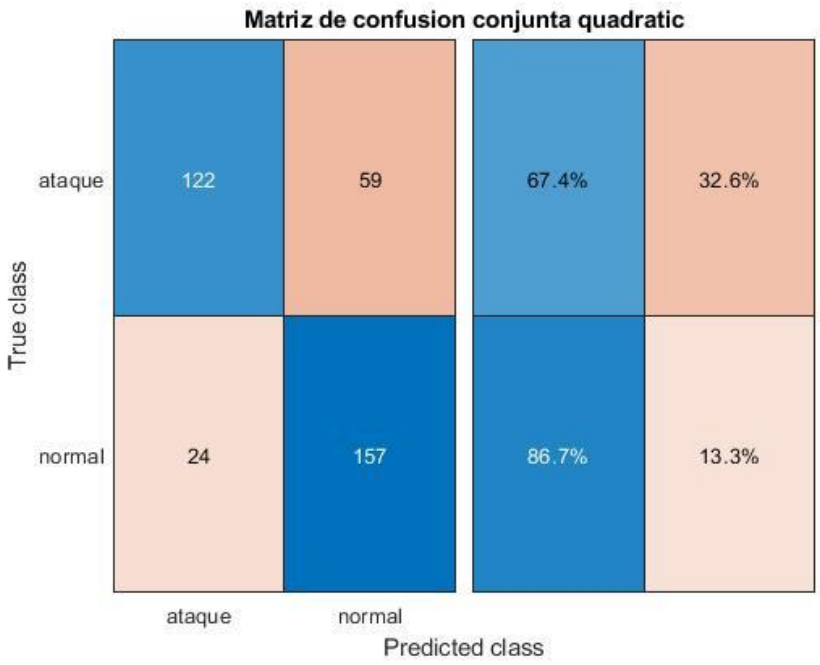


Figura 5-21: matriz de confusión conjunta quadratic

Dependiendo del tipo de condición que se aplique los resultados mejorarán para uno u otro tráfico. Si tenemos en cuenta la probabilidad de unión, se detecta con más probabilidad un ataque, aunque aumentaría el número de falsas alarmas, siendo esto último algo secundario. Si se tiene en cuenta que para clasificar un valor como ataque es necesario que en los dos espacios esté clasificado como ataque, la probabilidad de detectar un ataque disminuye mucho, siendo muy elevado el número de falsos negativos. No obstante, el tráfico normal se clasifica mucho mejor.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Partiendo de un TFG anterior que había llegado a la conclusión de que sí había diferencia de los parámetros α-estable entre el tráfico normal y de ataque se pensó que se podía hacer la segmentación de estos parámetros.

Gracias al autor del TFG de partida se pudo ahorrar mucho tiempo en desarrollar el código que hiciera posible el cálculo de los parámetros. Las series temporales también estaban creadas por lo que solo hubo que hacer unas pequeñas modificaciones para que Matlab calculase los parámetros α-estables a nuestra medida.

La forma en la que se decidió segmentar fue mediante la construcción de un espacio de fases formado por la combinación de 2 en 2 de los 4 parámetros α-estables. Previamente, se decidió que el tamaño de ventana ideal fuera de 5 minutos para no tener demasiados puntos que se solapasen entre sí.

Viendo el mal resultado que dio k-means se decidió ir por otras vías, como el aprendizaje supervisado, que estaba disponible en Matlab. Finalmente, aplicando el análisis discriminante se pudo llegar a la conclusión de que sí es posible una segmentación de estos parámetros para el tráfico normal y de ataque, con unas tasas de error bastante aceptables. Esto facilita el estudio de los ataques DoS, pudiendo establecer zonas de riesgo y detectar posibles amenazas a la red.

No es un método perfecto, pero este estudio puede ser la base para futuros trabajos donde se mejore esta clasificación mediante otro tipo de algoritmos y datos.

6.2 Trabajo futuro

De cara a un futuro trabajo, se podría realizar lo siguiente:

- Realizar un estudio de esto mismo, pero en diferentes entornos y con distintos datos para poder determinar la generalización de lo expuesto.

- Adaptar estos algoritmos e implementar un sistema que determine los parámetros α-estables, pero en tiempo real, basándose en los resultados de este estudio, de tal forma que se decidiese si ese valor es bueno o perjudicial para la red.

- Determinar la causa de la desviación de los parámetros α-estables cuando se produce un ataque DoS.

- Sistematización de pruebas futuras.

Referencias

[1] Eric Crusi Mozota. “Estudio de ciberataques mediante el análisis de tráfico en Internet.”

[2] Ejemplo de distribución alfa estable. https://commons.wikimedia.org/wiki/File:Levy_distributionPDF.pn

[3] Jhon P. Nolan. “Stable Distributions”

[4] Función *classify()*. https://es.mathworks.com/help/stats/examples/classification.html

[5] Función *fitcdiscr()*. https://es.mathworks.com/help/stats/fitcdiscr.html?s_tid=doc_ta

[6] Luis de Pedro, Eric Crusi, Alberto Ruiz, Jorge E. López de Vergara. “Detección de ciberataques mediante análisis estadístico con distribuciones α-estables”

[7] Ataque de denegación de servicio. https://es.wikipedia.org/wiki/Ataque_de_denegaci%C3%B3n_de_servicio

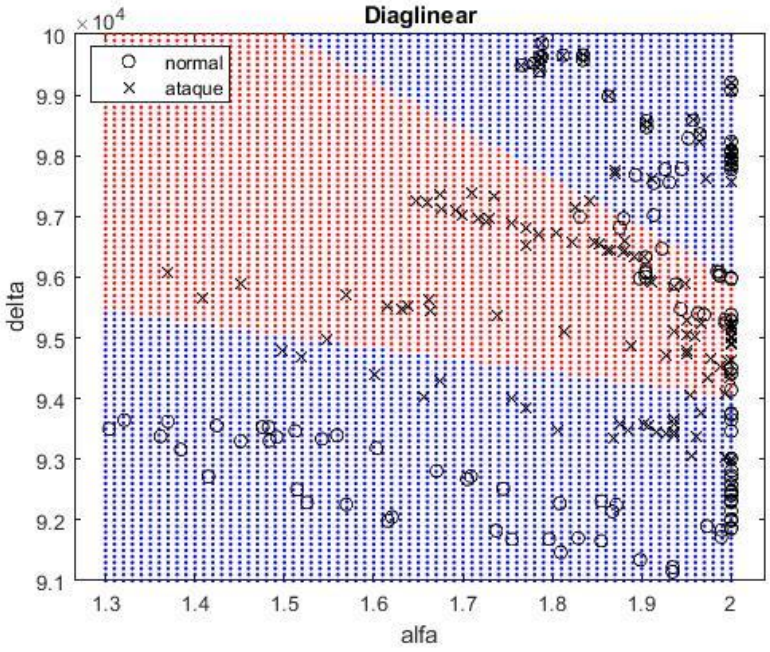
Glosario

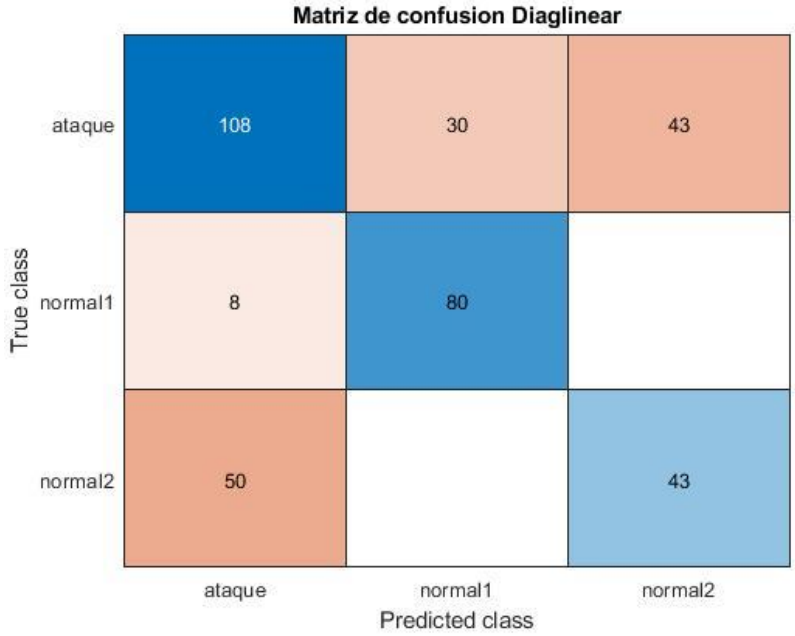
IP	Internet Protocol
AWK	Lenguaje de programación
DoS	Denegación de Servicio
DDoS	Denegación de Servicio Distribuida
SYN	Flag de sincronización del protocolo de transporte de Internet (TCP)
ICMP	Protocolo de mensajes de control de Internet
SMURF	Variante de ataque por inundación ICMP
ACK	Flag de asentimiento del protocolo de transporte de Internet (TCP)

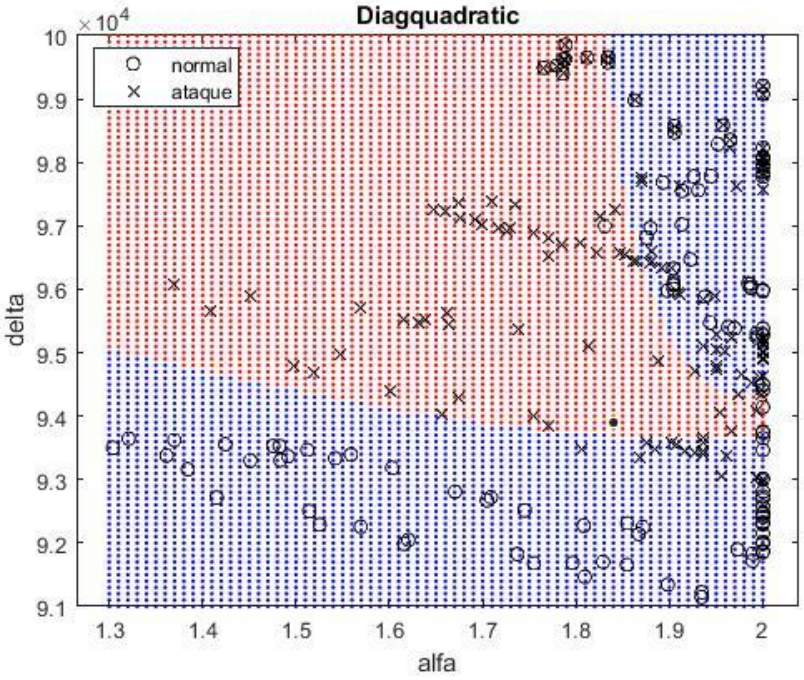
Anexos

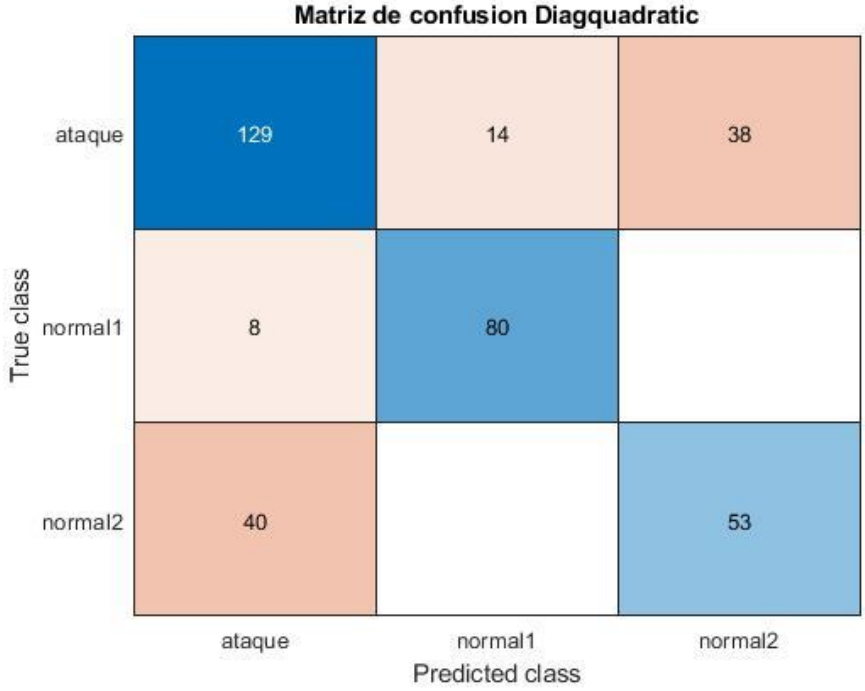
A Gráficas adicionales

En este anexo se muestran todas las gráficas que se han ido realizando a lo largo del trabajo. Esto no significa que sean gráficas que no tengan ningún valor, pero debido al elevado número de estas se ha decidido que las menos importantes vayan en esta sección a pesar de que hayan sido útiles para el desarrollo del trabajo.

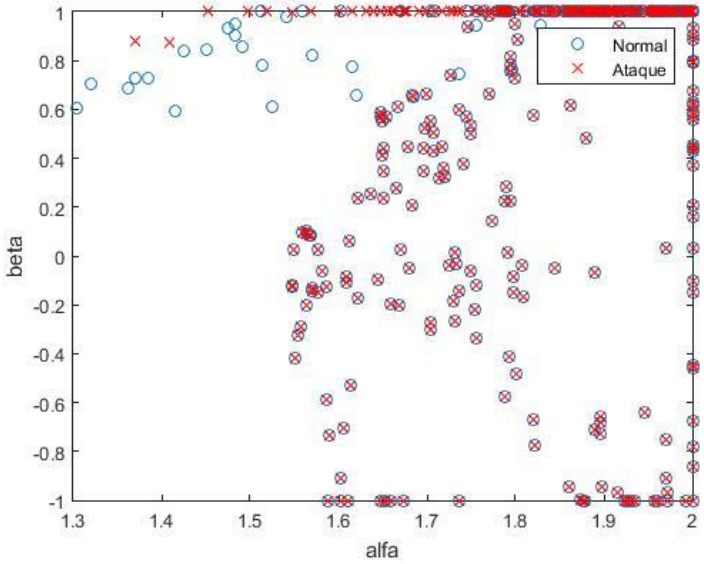


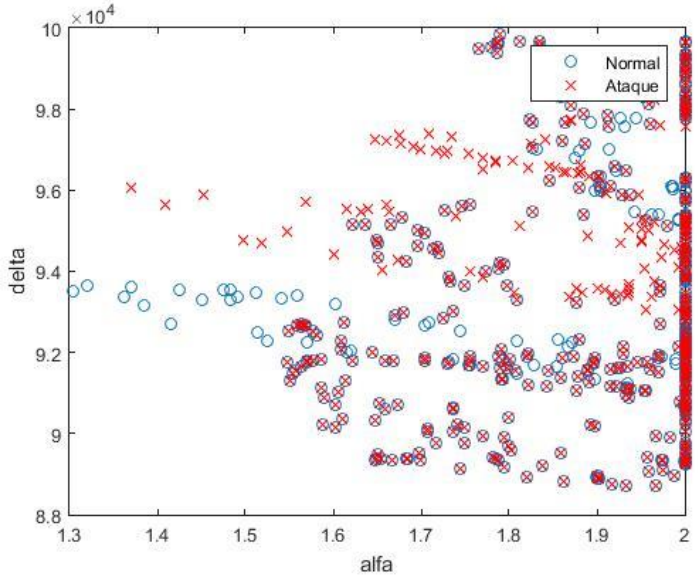


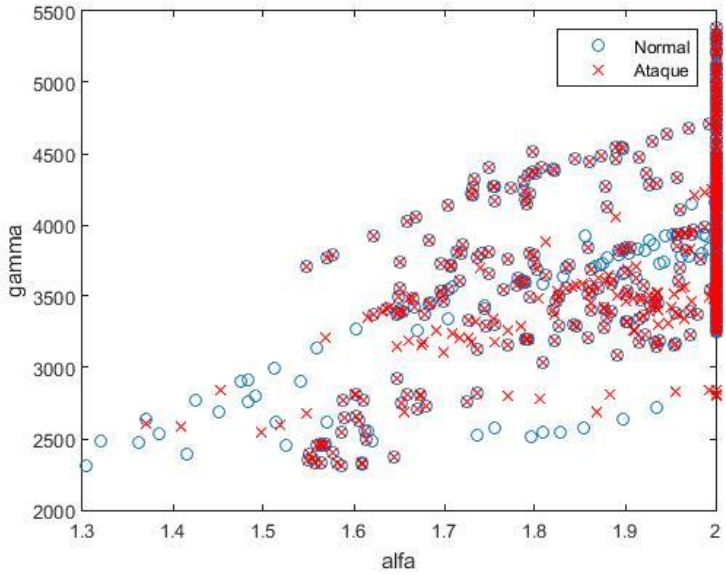

Diagquadratic
×10^4
10
9.9
9.8
9.7
9.6
9.5
9.4
9.3
9.2
9.1
delta
normal
ataque
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
alfa

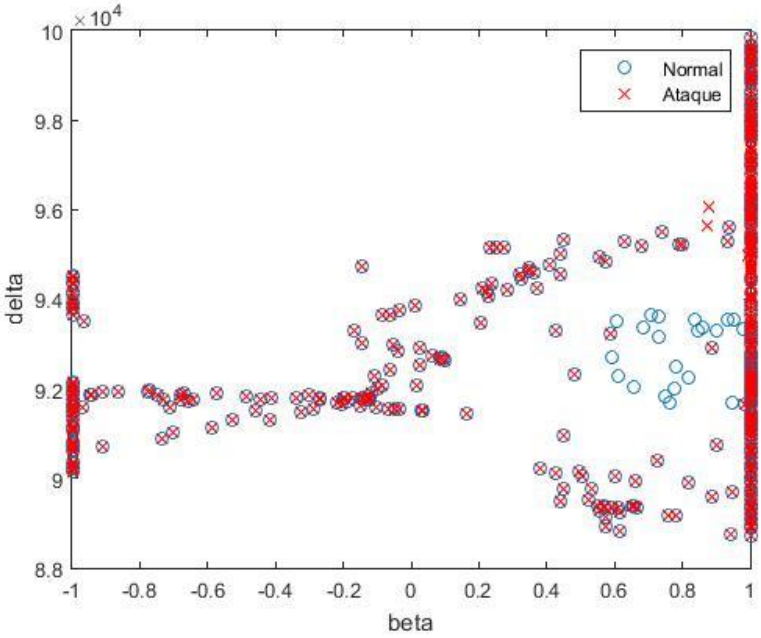

Matriz de confusion Diagquadratic
ataque
normal1
normal2
True class
129
14
38
8
80
40
53
ataque
normal1
normal2
Predicted class

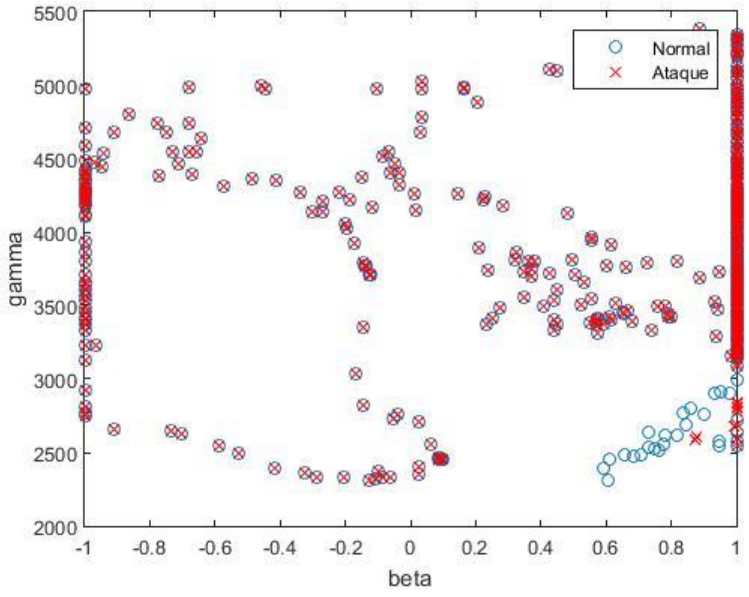
Espacios de fases para ventana de 15 minutos



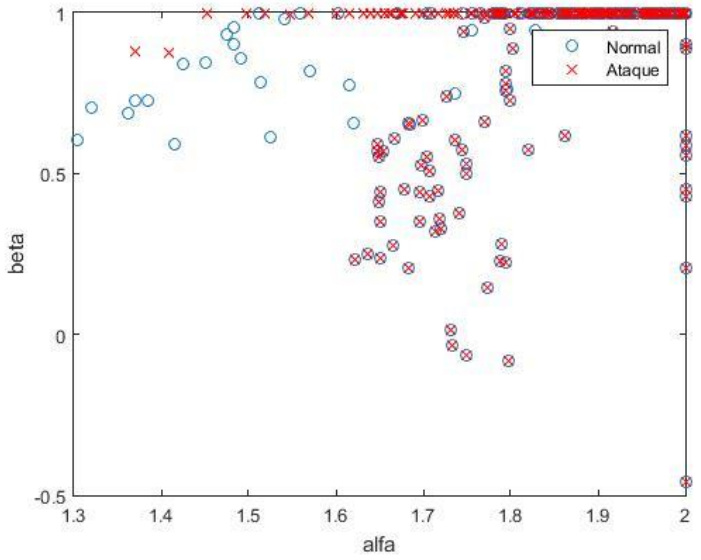


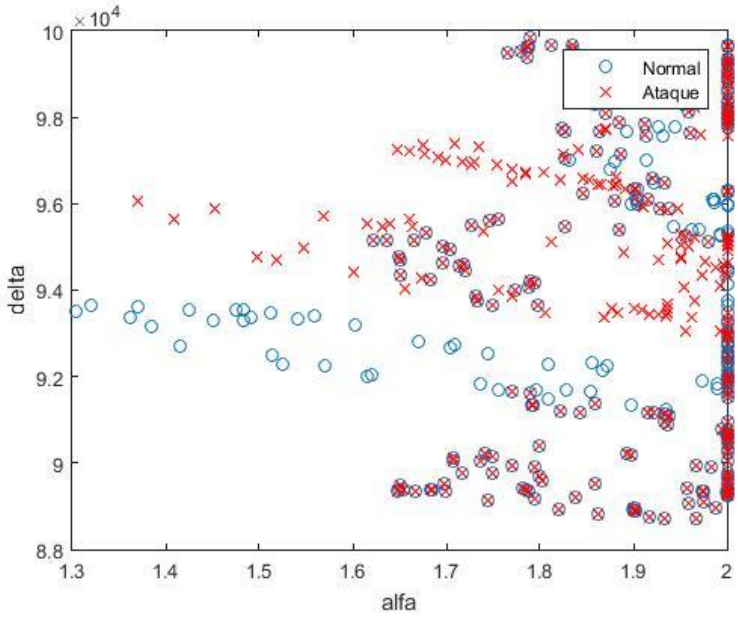


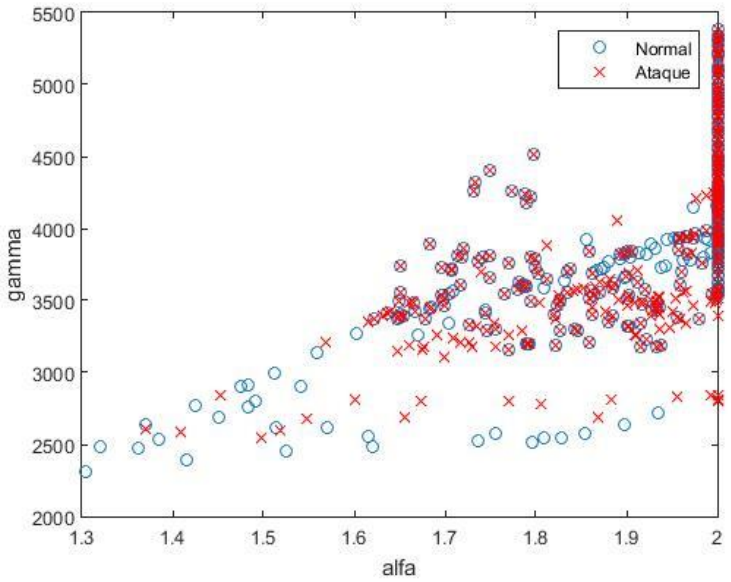

×10^4
10
9.8
9.6
9.4
9.2
9
8.8
delta
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
beta
Normal
Ataque

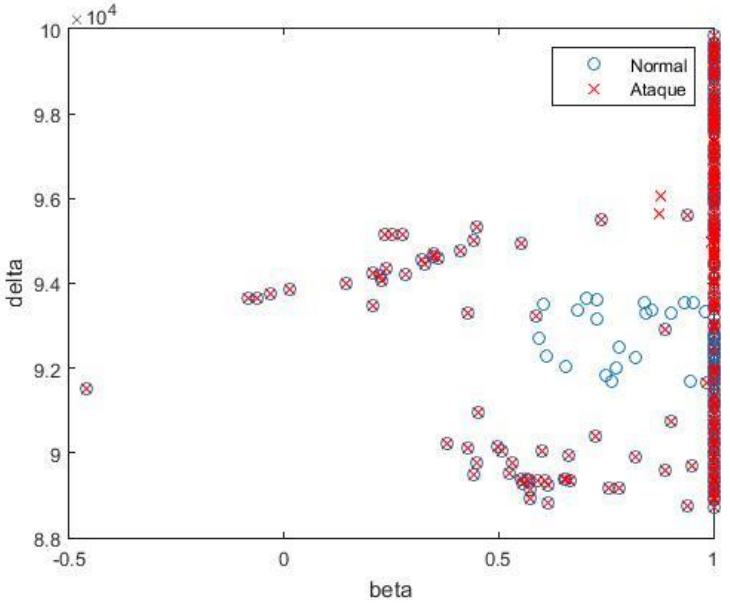

5500
5000
4500
4000
3500
3000
2500
2000
gamma
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
beta
Normal
Ataque

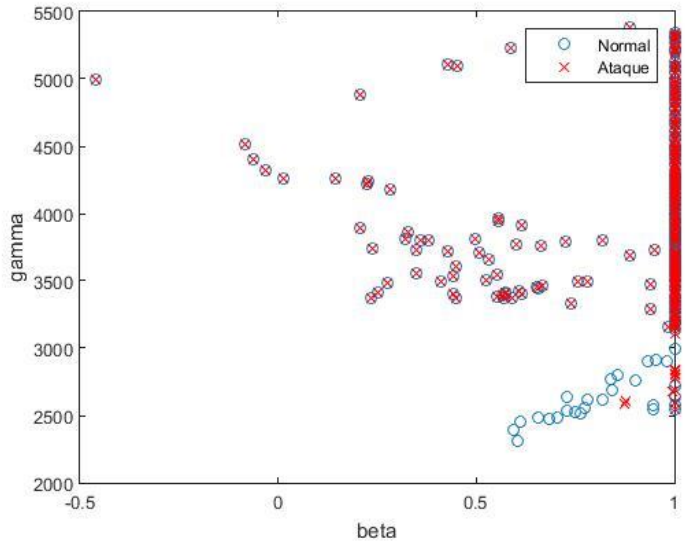
Espacios de fases para ventana de 10 minutos

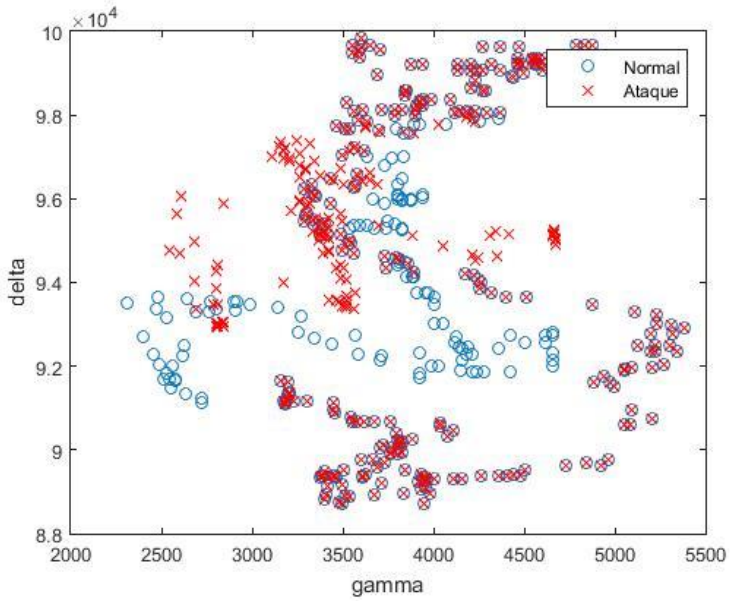




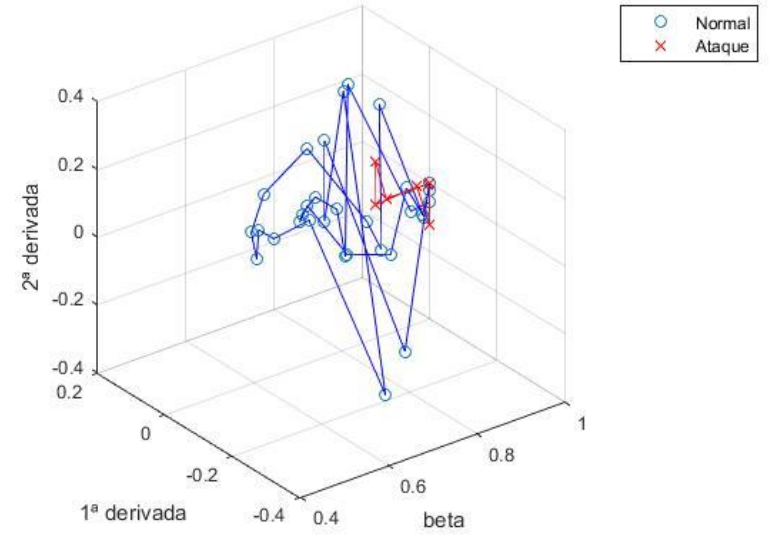


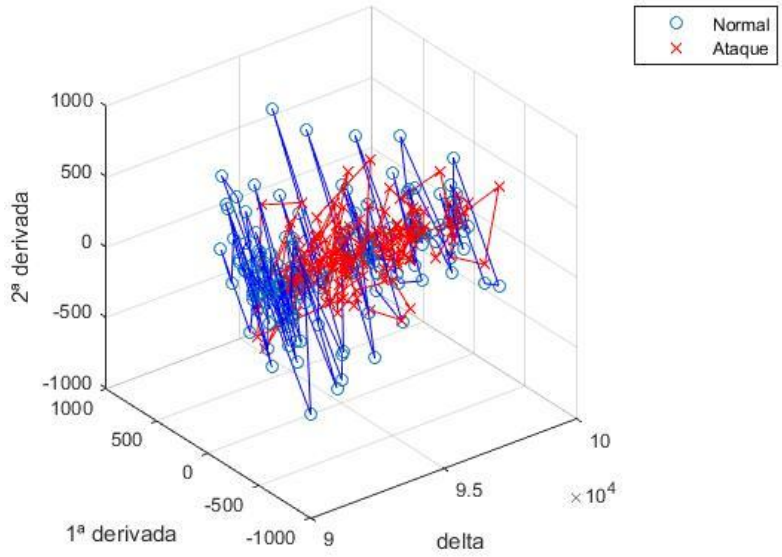

×10^4
10
9.8
9.6
9.4
9.2
9
8.8
delta
-0.5
0
0.5
1
beta
Normal
Ataque

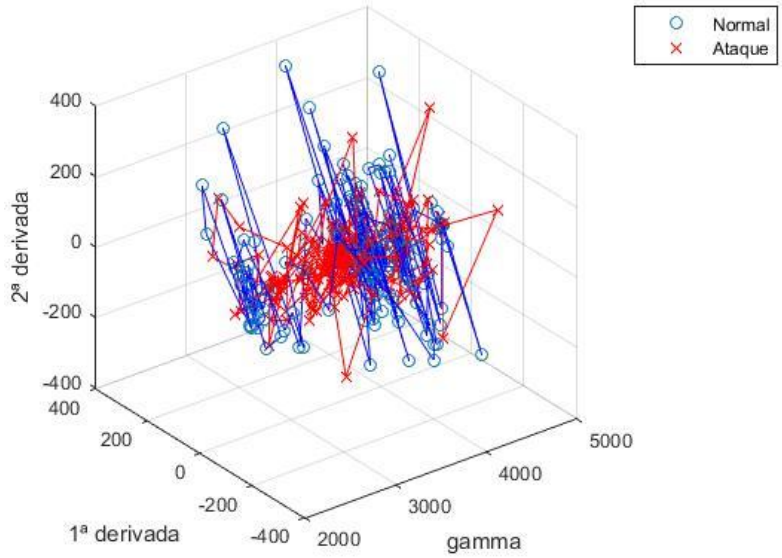

5500
5000
4500
4000
3500
3000
2500
2000
gamma
-0.5
0
0.5
1
beta
Normal
Ataque


×10^4
10
9.8
9.6
9.4
9.2
9
8.8
delta
2000
2500
3000
3500
4000
4500
5000
5500
gamma
Normal
Ataque

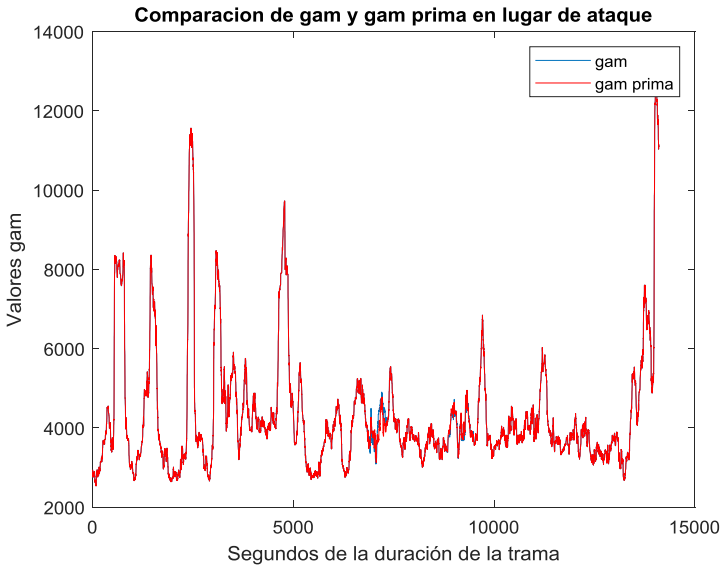
Derivadas en 3D

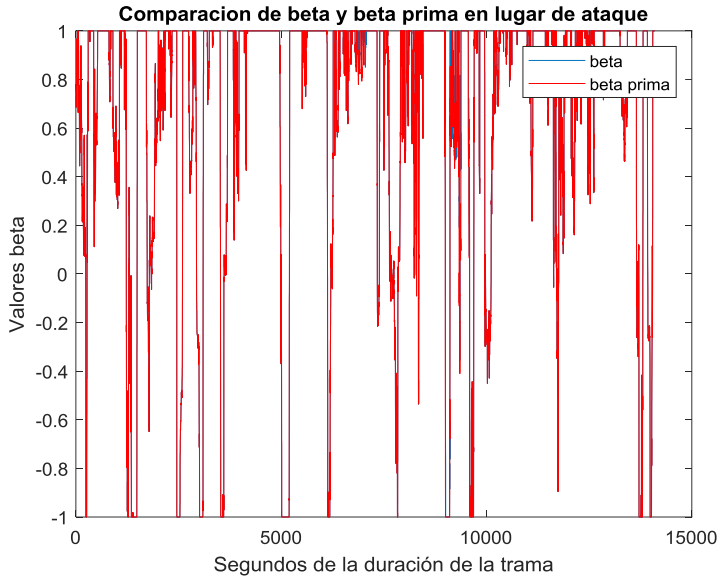


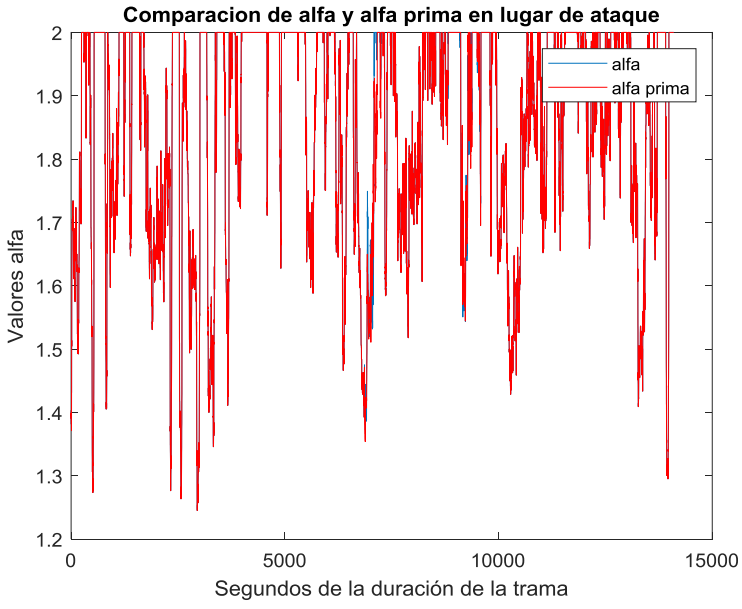




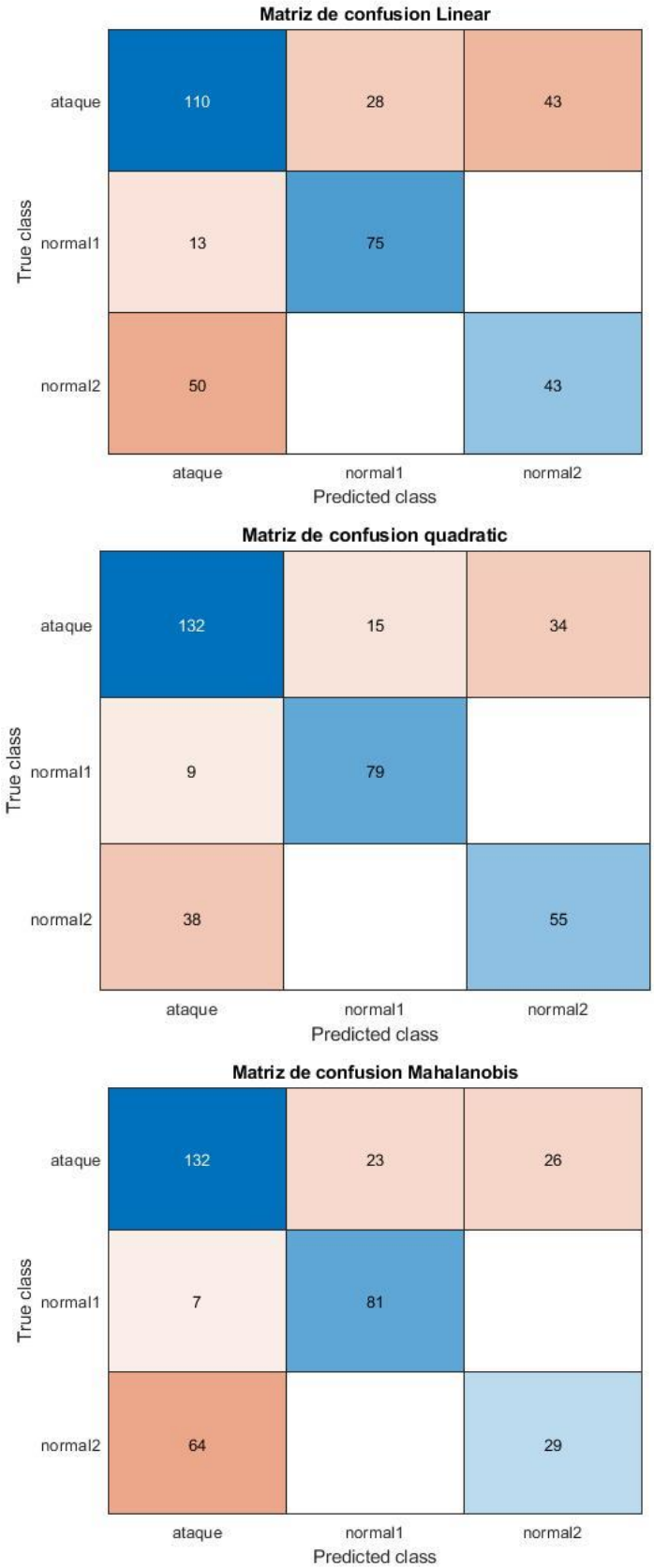
Parámetros α-estables ventana de 5 minutos







Matrices de confusión (γ, δ)



Matrices de confusión (α, δ)